

A Two-Level Approach to Large Mixed-Integer Programs with Application to Cogeneration in Energy-Efficient Buildings*

Fu Lin, Sven Leyffer, and Todd Munson[†]

April 20, 2015

Abstract

We study a two-stage mixed-integer linear program (MILP) with more than 1 million binary variables in the second stage. We develop a two-level approach by constructing a semi-coarse model (coarsened with respect to variables) and a coarse model (coarsened with respect to both variables and constraints). We coarsen binary variables by selecting a small number of pre-specified daily on/off profiles. We aggregate constraints by partitioning them into groups and summing over each group. With an appropriate choice of coarsened profiles, the semi-coarse model is guaranteed to find a feasible solution of the original problem and hence provides an upper bound on the optimal solution. We show that solving a sequence of coarse models converges to the same upper bound with proven finite steps. This is achieved by adding violated constraints to coarse models until all constraints in the semi-coarse model are satisfied. We demonstrate the effectiveness of our approach in cogeneration for buildings. The coarsened models allow us to obtain good approximate solutions at a fraction of the time required by solving the original problem. Extensive numerical experiments show that the two-level approach scales to large problems that are beyond the capacity of state-of-the-art commercial MILP solvers.

Keywords: Coarsened models, distributed generation, large-scale problems, two-level approach, multi-period planning, resource and cost allocation, two-stage mixed-integer programs.

AMS subject classifications: 91B32, 90C06, 90C11, 90C90.

1 Problem Definition and Motivation

We consider a hierarchical two-stage mixed-integer linear program (MILP) with integer variables in both the first and second stages. We are particularly interested in applications where the first-stage integer variables model design or purchasing decisions and the second-stage variables model operational decisions over a long time horizon (e.g., hourly operations over a decadal horizon). The goal is to take operational constraints into account when making a capital investment decision. Thus, our model is complicated by the fact that second-stage variables include both binary (on/off) decisions and continuous variables that model operational settings. Examples include the design of cogeneration units for commercial buildings subject to operational conditions (Siddiqui et al., 2005; Stadler et al., 2009; Ren and Gao, 2010; Pruitt et al., 2013, 2014) and transmission network expansion subject to unit commitment constraints (Alguacil et al., 2003; Hedman et al., 2010;

*Preprint ANL/MCS-P5332-0415

[†]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 (fulin@mcs.anl.gov, leyffer@mcs.anl.gov, tmunson@mcs.anl.gov).

Munoz et al., 2013, 2014). Models of this class can involve hundreds of thousands or even millions of binary variables and are beyond the scope of today's state-of-the-art solvers.

We consider a two-stage MILP with m first-stage variables and three sets of N second-stage variables. The first-stage variables are $y \in \{0, 1\}^m$. We have three classes of second-stage variables: (1) on/off decisions, $x \in \{0, 1\}^N$; (2) operational settings $v \in \mathbb{R}^N$ that are switched on or off by x , that is, $Lx \leq v \leq Ux$ for finite bounds $L \leq U$; and (3) other second-stage variables $0 \leq w \in \mathbb{R}^N$. The MILP model is described by

$$\begin{aligned} & \underset{y, x, v, w}{\text{minimize}} && a^T y + b^T x + c^T v + d^T w \\ & \text{subject to} && Ay + Bx + Cv + Dw \leq f \\ & && y \in \{0, 1\}^m \\ & && x \in \{0, 1\}^N \\ & && v \in \mathbb{R}^N, \quad Lx \leq v \leq Ux \\ & && w \in \mathbb{R}^N, \quad w \geq 0, \end{aligned} \tag{1.1}$$

where $a \in \mathbb{R}^m$, $b, c, d \in \mathbb{R}^N$, $A \in \mathbb{R}^{M \times m}$, and $B, C, D \in \mathbb{R}^{M \times N}$. We assume that $m \ll N$ and $m \ll M$; that is, the number of first-stage variables is much smaller than the number of second-stage binary variables and coupling constraints.

In addition to a large number of binary and continuous variables in the second stage, the challenges of model (1.1) also arise from coupling constraints. We do not assume any sparsity structure in matrices B , C , and D . This is in contrast to the arrow-type sparsity structure that arises, for example, in stochastic programs (Birge and Louveaux, 2011). Therefore, fixing the first-stage variable y does not decompose (1.1) into scenario-based subproblems. Moreover, the coupling constraints in (1.1) are not amenable to decomposition methods such as Lagrangian relaxation (Geoffrion, 1974; Fisher, 1981). The reason is that the number of coupling constraints, M , is of the same order as the number of variables, N . Dualizing all coupling constraints results in a large number of dual variables; hence, Lagrangian relaxation of (1.1) does not yield efficient decomposition methods (Fisher, 1981).

Problem (1.1) arises in several applications. In particular, our work is motivated by the cogeneration problem with renewable energy for commercial buildings. In this case, the first-stage design involves investment decisions for cogeneration units such as fuel cells, solar panels, and battery storage. The second-stage problem aims at optimal on/off hourly operation that takes into account technology specifics such as minimum/maximum power generation. Our goal is to include operational constraints in the design of cogeneration.

One of our objectives is to find the optimal first-stage solution of (1.1). However, the two-stage MILP with a large number of binary variables at the second stage is beyond the scope of state-of-the-art commercial MIP solvers. For example, a typical cogeneration model with a ten-year horizon results in 1.05 million binary variables (3650 days \times 24 hours \times 12 units). On the other hand, a naive approach that solves (1.1) with a short horizon at the second stage provides first-stage designs that are suboptimal for a long horizon problem. The reason is that short horizon problems do not take into account coupling constraints over a long horizon. Moreover, the problem data of short-horizon problems are not representative of long-horizon problems, resulting in suboptimal solutions.

We develop a *two-level approach* that coarsens the hourly on/off variables to daily operation profiles. Since the profile representation yields a model with many fewer variables, we refer to this step as the *primal (variable) coarsening*. The resulting *semi-coarse model* still contains the same order of constraints as in (1.1). We reduce the number of constraints by partitioning them into groups that are of the same size as the profiles and summing over each group. This aggregation of

constraints results in a relaxed problem whose solutions may not be feasible for the original MILP model. We include the violated constraints, re-solve the coarsened MILP model, and repeat this process until all constraints are satisfied. We refer to the aggregation of constraints as the *dual (constraint) coarsening* and the resulting MILP as the *coarse model*.

1.1 Literature Review

The idea of aggregating variables and constraints to build approximate optimization models is not new. Zipkin studies the effect of variable aggregation and row aggregation for linear programs in (Zipkin, 1980b,a). This framework is extended to stochastic linear programs by Birge (1985) and by Clay and Grossmann (1997). For integer programs, a large amount of work focuses on aggregating constraints into one or more *surrogate* constraints for which one can show that the solution of the original problem and that of the surrogate problem are identical. Early work includes that of Balas (1965), Glover (1968), Geoffrion (1969), and Glover (1977); see also the survey paper by Rogers et al. (1991).

We note that the theory of surrogate constraints focuses mainly on integer programs with non-negative integer coefficients. The seminal work by Mathews (1896) results in an exponential grow of the coefficients in the surrogate constraints. Many refinements have been developed, including simultaneous and sequential aggregation schemes, to reduce the magnitude of coefficients (Rogers et al., 1991). For mixed-integer programs with real-valued coefficients, however, we are not aware of aggregation schemes that guarantee equivalence between the original problem and the surrogate problem. In our two-level approach, we employ constraint aggregation as a relaxation technique to identify a smaller set of constraints that are active at the optimal solution. This is achieved by solving a sequence of MILPs and adding the violated constraints until all constraints in the original MILP are satisfied. Furthermore, we take advantage of the LP warm-start to reduce the number of MILP re-solves and the computational time of each MILP.

A large body of literature appeared in the 1990s on bilevel or multilevel mixed integer programs; see (Moore and Bard, 1990; Wen and Yang, 1990; Edmunds and Bard, 1992; Hansen et al., 1992; Wen and Huang, 1996) and survey papers (Wen and Hsu, 1991; Vicente and Calamai, 1994). One of the main theoretical emphases is on decentralized decision-making from a game-theoretic point of view (Wen and Hsu, 1991). Optimality conditions for convex bilevel programs have been established in (Vicente and Calamai, 1994). A heuristic-based branch-and-bound method is developed by Wen and Yang (1990) and Hansen et al. (1992), and tabu search method is introduced by Wen and Huang (1996). Recent years have seen specific application-driven algorithms ranging from infrastructure protection planning (Scaparra and Church, 2008) to vulnerability analysis of power grids (Pinar et al., 2010).

In this context, the multilevel method is specifically referred to the problem formulation. In contrast, our two-level method concentrates on algorithmic development for mixed-integer linear programs. In particular, our two-level approach builds optimization models of different resolutions from a fine-level problem to a coarse-level problem. We develop a systematic procedure that coarsens binary and continuous variables in addition to the aggregation of constraints. Our approach allows us to solve large MILPs with more than one million binary and continuous variables and coupling constraints. Extensive numerical experiments have been conducted to verify the efficiency of the developed algorithm.

Our two-level approach is reminiscent of multi-grid methods in linear algebra and solution of partial differential equations. Related work includes that of Gelman and Mandel (1990), who discussed a multilevel iterative method for generic optimization problem; Gratton et al. (2008), who developed a recursive trust-region method for nonlinear unconstrained problems; and Wen

and Goldfarb (2009), who proposed a line search multi-grid approach for nonlinear programs.

1.2 Our Contributions

Our contributions are fourfold. First, we develop a systematic two-level approach for two-stage MILPs with a large number of binary and continuous variables in the second stage. The coarsening of binary variables is done by introducing profiles (vectors) with binary elements. By selecting one profile from a small number of candidate profiles, we reduce the number of binary variables by orders of magnitude. In addition, we reduce the number of continuous variables by using a convex combination of profiles with real elements. We show that the coarsening in variables results in a tightening of the original MILP and therefore provides an upper bound on the optimal solution.

Second, we devise a simple scheme for constraint aggregation that does not require any sparsity assumptions on the coefficient matrices. By partitioning constraints into groups and summing up constraints in each group, we obtain a relaxation of the original MILP with many fewer constraints. We solve a sequence of MILPs by adding any violated constraints until all constraints are satisfied. We show that the LP-relaxation of MILPs can be used to warm-start the MILP re-solves and significantly reduce the computational effort.

Third, we implement our two-level approach in the context of the design of cogeneration for buildings. The resulting complex MILP model has a large number of coupling constraints over a long time horizon. We employ a moving-horizon method to generate valid profiles. We show by construction that the generated profiles satisfy coupling constraints in time, hence further reducing the number of constraints in the coarsened models.

Fourth, we verify the efficiency of our algorithm on a variety of examples generated from simulation programs for commercial buildings. While our two-level approach requires solving a sequence of coarse MILPs, numerical results indicate that the first iterate provides good approximate solutions of the semi-coarse models. Through extensive numerical experiments, we demonstrate the scalability of the two-level approach on a rich set of large problems that are beyond the capacity of state-of-the-art commercial solvers.

Outline. The remainder of this paper is organized as follows. In Section 2, we describe the two-level approach for the two-stage MILPs. We show that our approach results in a tightening of the original problem and provides an upper bound on the optimal solution. In Section 3, we apply our approach to a complex MILP model from the cogeneration problem in buildings. We discuss how the profiles are generated and selected such that the coarsened models provide feasible solutions to (1.1). In Section 4, we demonstrate the effectiveness of our approach on a diverse set of test problems. We show that the two-level approach allows us to find good approximate solutions with a fraction of computational time compared with solving the full model. In Section 5, we summarize our contribution and discuss future extensions.

2 Two-Level Approach to MILP

In this section, we derive two models that are formulated in terms of the first-stage variables and $n \ll N$ second-stage variables. We start by presenting our ideas for variable coarsening, resulting in a semi-coarse model that has $\mathcal{O}(n)$ second-stage variables and $\mathcal{O}(N)$ second-stage constraints. Next we show how to further coarsen this model by aggregating constraints, resulting in a coarse model with both $\mathcal{O}(n)$ second-stage variables and constraints.

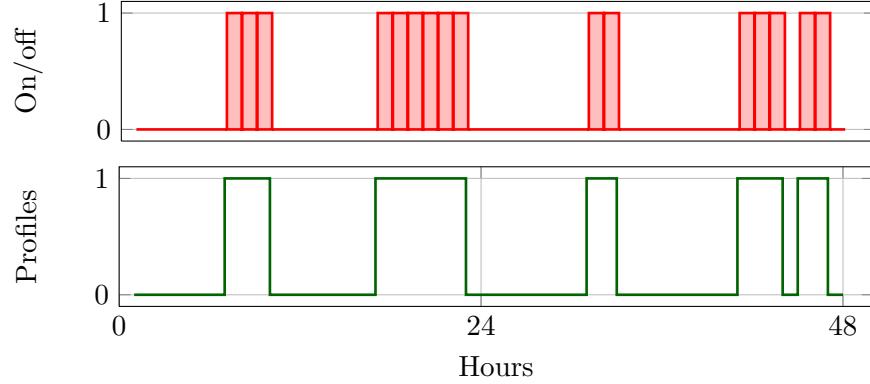


Figure 1: The first row shows the hourly on/off variables $x_t \in \{0, 1\}$ for 48 hours. The second row shows the profile representation with two daily profiles $\bar{X}_k \in \{0, 1\}^{24}$.

2.1 Primal Coarsening

To coarsen the variables in the model, we introduce a coarsening factor $\delta \in \mathbb{Z}_+$ and group N second-stage variables into $n = N/\delta$ equal-sized groups of size δ (the assumption that the groups are equal-sized is not critical):

$$x_i = \begin{pmatrix} x_{\delta(i-1)+1} \\ \vdots \\ x_{\delta i} \end{pmatrix} \quad \text{for } i = 1, \dots, n. \quad (2.2)$$

We define groups for v and w analogously. These groups correspond to a partition of the second-stage variables x, v , and w :

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad v = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}, \quad w = \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}. \quad (2.3)$$

Next, we introduce δ -profiles, which are fixed parameter values for a group of variables:

$$\bar{X}_k \in \{0, 1\}^\delta \quad \text{for } k = 1, \dots, K. \quad (2.4)$$

We show in Section 3 how these profiles are generated and selected. Figure 1 illustrates that $N = 48$ hourly on/off variables can be represented by $n = 2$ daily profiles with length $\delta = 24$. Now for every \bar{X}_k , we collect a set of I_k operational δ -profiles for \bar{V} and \bar{W} :

$$\bar{V}_{jk} \in \mathbb{R}^\delta, \quad \text{with } L\bar{X}_k \leq \bar{V}_{jk} \leq U\bar{X}_k \quad \text{for } j = 1, \dots, I_k, \quad k = 1, \dots, K, \quad (2.5)$$

and

$$0 \leq \bar{W}_j \in \mathbb{R}^\delta \quad \text{for } j = 1, \dots, J, \quad (2.6)$$

respectively.

Given these sets of δ -profiles, we perform a change of variables that replaces the second-stage variables (x, v, w) by a reduced set of variables $(\bar{x}, \bar{v}, \bar{w})$ for the δ -profiles, resulting in a coarsening of the second-stage variables. In particular, we set

$$x_i = \sum_{k=1}^K \bar{x}_{ik} \bar{X}_k, \quad \sum_{k=1}^K \bar{x}_{ik} \leq 1, \quad \bar{x}_{ik} \in \{0, 1\} \quad \text{for } i = 1, \dots, n, \quad k = 1, \dots, K. \quad (2.7)$$

Thus, we have that $\bar{x} \in \{0, 1\}^{Kn}$, and our goal is to create models where $Kn \ll N$. Similarly, we write v and w as

$$v_i = \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \bar{V}_{jk}, \quad \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \leq 1, \quad \sum_{j=1}^{I_k} \bar{v}_{ijk} = \bar{x}_{ik}, \quad 0 \leq \bar{v}_{ijk} \leq 1, \quad \text{for } i = 1, \dots, n, \quad (2.8)$$

and

$$w_i = \sum_{j=1}^J \bar{w}_{ij} \bar{W}_j, \quad \sum_{j=1}^J \bar{w}_{ij} \leq 1, \quad \text{for } i = 1, \dots, n. \quad (2.9)$$

We note that we do not enforce $\bar{w}_{ij} \in \{0, 1\}$ or $\bar{v}_{ijk} \in \{0, 1\}$. The reason is that we wish to allow more freedom for choosing operational profiles in the second stage as long as they remain feasible. This choice also simplifies the coarsened second-stage problem and provides a valid approach if we assume that convex combinations of operational δ -profiles \bar{V}_{jk} and \bar{W}_j are feasible, which is often the case in practice.

The partition of variables (x, v, w) implies a partition of the problem matrices B , C , and D of (1.1) as

$$B = [B_1 : \dots : B_n], \quad C = [C_1 : \dots : C_n], \quad D = [D_1 : \dots : D_n],$$

where B_i , C_i , and $D_i \in \mathbb{R}^{M \times \delta}$ for $i = 1, \dots, n$. Next, we define δ -profile matrices as

$$\bar{X} = [\bar{X}_1 : \dots : \bar{X}_K], \quad \bar{V} = [\bar{V}_{11} : \dots : \bar{V}_{1I_1} : \dots : \bar{V}_{K1} : \dots : \bar{V}_{KI_K}], \quad \bar{W} = [\bar{W}_1 : \dots : \bar{W}_J], \quad (2.10)$$

where \bar{X}_k , \bar{V}_{jk} , and \bar{W}_j are vectors of length δ . We define aggregated (coarse) matrices \bar{B} , \bar{C} , and $\bar{D} \in \mathbb{R}^{M \times n\delta}$ as

$$\bar{B} = [B_1 \bar{X} : \dots : B_n \bar{X}], \quad \bar{C} = [C_1 \bar{V} : \dots : C_n \bar{V}], \quad \bar{D} = [D_1 \bar{W} : \dots : D_n \bar{W}], \quad (2.11)$$

and aggregated cost vectors $\bar{b}, \bar{c}, \bar{d} \in \mathbb{R}^{Kn}$ as

$$\bar{b} = [b_1^T \bar{X} : \dots : b_n^T \bar{X}]^T, \quad \bar{c} = [c_1^T \bar{V} : \dots : c_n^T \bar{V}]^T, \quad \bar{d} = [d_1^T \bar{W} : \dots : d_n^T \bar{W}]^T. \quad (2.12)$$

We obtain the following aggregated MILP, which we refer to as the *semi-coarse MILP*, because it has been coarsened in primal variables only:

$$\begin{aligned} & \underset{y, \bar{x}, \bar{v}, \bar{w}}{\text{minimize}} && a^T y + \bar{b}^T \bar{x} + \bar{c}^T \bar{v} + \bar{d}^T \bar{w} \\ & \text{subject to} && A y + \bar{B} \bar{x} + \bar{C} \bar{v} + \bar{D} \bar{w} \leq f \\ & && \sum_{k=1}^K \bar{x}_{ik} \leq 1, \quad \bar{x}_{ik} \in \{0, 1\} \\ & && \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \leq 1, \quad \sum_{j=1}^{I_k} \bar{v}_{ijk} = \bar{x}_{ik}, \quad 0 \leq \bar{v}_{ijk} \leq 1 \\ & && \sum_{j=1}^J \bar{w}_{ij} \leq 1, \quad 0 \leq \bar{w}_{ij} \leq 1. \end{aligned} \quad (2.13)$$

The MILP (2.13) has potentially many fewer variables than (1.1), but contains as many constraints as the original problem. Before we show how to aggregate constraints in the next section, we finish this section by summarizing some of the properties of (2.13).

Proposition 2.1. *Let $(\bar{x}, \bar{v}, \bar{w})$ be a feasible point of the semi-coarse model (2.13), then it follows that the corresponding fine-scale variables*

$$x = \begin{pmatrix} \sum_{k=1}^K \bar{x}_{1k} \bar{X}_k \\ \vdots \\ \sum_{k=1}^K \bar{x}_{nk} \bar{X}_k \end{pmatrix}, \quad v = \begin{pmatrix} \sum_{k,j=1}^{K,I_k} \bar{v}_{1jk} \bar{V}_{jk} \\ \vdots \\ \sum_{k,j=1}^{K,I_k} \bar{v}_{njk} \bar{V}_{jk} \end{pmatrix}, \quad w = \begin{pmatrix} \sum_{j=1}^J \bar{w}_{1j} \bar{W}_j \\ \vdots \\ \sum_{j=1}^J \bar{w}_{nj} \bar{W}_j \end{pmatrix} \quad (2.14)$$

are feasible in the original MILP (1.1).

Proof. The binary constraint $x \in \{0, 1\}^N$ follows from the representation (2.7) and the definition of binary vectors \bar{X}_k . Similarly, the non-negativity of w is a direct consequence of non-negative profiles \bar{W}_j and non-negative coefficients \bar{w}_{ij} . To show that v is feasible, we start with the definition of \bar{V}_{jk} in (2.5):

$$L \cdot \bar{X}_k \leq \bar{V}_{jk} \leq U \cdot \bar{X}_k.$$

Since $\bar{v}_{ijk} \geq 0$, it follows that

$$L \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \bar{X}_k \leq \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \bar{V}_{jk} \leq U \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \bar{X}_k.$$

Using $\bar{x}_{ik} = \sum_{j=1}^{I_k} \bar{v}_{ijk}$, we have

$$L \sum_{k=1}^K \bar{x}_{ik} \bar{X}_k \leq v_i \leq U \sum_{k=1}^K \bar{x}_{ik} \bar{X}_k, \quad \text{for } i = 1, \dots, n.$$

Using the definition of x_i in (2.14), it follows that $Lx_i \leq v_i \leq Ux_i$, for $i = 1, \dots, n$, and thus $Lx \leq v \leq Ux$. The proof is complete by noting that

$$f \geq Ay + \bar{B}\bar{x} + \bar{C}\bar{v} + \bar{D}\bar{w} = Ay + Bx + Cv + Dw,$$

where the equality follows from the definition of aggregated matrices (2.11) and the representation of the fine-scale variables (2.14). \square

Next, we show that the semi-coarse model provides an upper bound.

Proposition 2.2. *The semi-coarse model (2.13) is a tightening of the original MILP (1.1), and its solution provides an upper bound on (1.1). The two problems are equivalent if the optimal profiles from the solution of (1.1) are included in (2.13).*

Proof. Let z_{MILP}^* and z_{semi}^* be the optimal value of the original MILP (1.1) and the semi-coarse model (2.13), respectively. Since any feasible point of the semi-coarse model (2.13) is a feasible point of the original MILP (1.1), it follows that (2.13) is a tightening of (1.1) and thus

$$z_{\text{MILP}}^* \leq z_{\text{semi}}^*.$$

Let (x^*, v^*, w^*) be the optimal solution of MILP (1.1). We now extract optimal profiles $(\bar{X}^*, \bar{V}^*, \bar{W}^*)$ corresponding to (x^*, v^*, w^*) . Then it follows that there exists a solution of (2.13) such that

$$x_i^* = \sum_{k=1}^K \bar{x}_{ik}^* \bar{X}_k^*, \quad v_i^* = \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk}^* \bar{V}_{jk}^*, \quad w_i^* = \sum_{j=1}^J \bar{w}_{ij}^* \bar{W}_j^*, \quad \text{for } i = 1, \dots, n.$$

Moreover, solution $(\bar{x}^*, \bar{v}^*, \bar{w}^*)$ is feasible in (2.13), and the objective value is the same as solution (x^*, v^*, w^*) . Because (2.13) is a tightening of (1.1), there cannot be a better solution than $(\bar{x}^*, \bar{v}^*, \bar{w}^*)$ that we constructed. Hence, $(\bar{x}^*, \bar{v}^*, \bar{w}^*)$ is the optimal solution, and both problems (1.1) and (2.13) are equivalent. \square

2.2 Dual Coarsening

We next coarsen the constraints by partitioning them into n groups of size δ and summing over each group. We define the rows of A as A_i^T , and we define a set of aggregated constraints by summing over the rows of A within each group:

$$\hat{A} := \begin{pmatrix} \sum_{i=1}^{\delta} A_i^T \\ \vdots \\ \sum_{i=1}^{\delta} A_{\delta+i}^T \\ \vdots \\ \sum_{i=1}^{\delta} A_{\delta(n-1)+i}^T \end{pmatrix}.$$

We aggregate the constraint matrices \bar{B} , \bar{C} , and \bar{D} in a similar way

$$\hat{\bar{B}} := \begin{pmatrix} \sum_{i=1}^{\delta} \bar{B}_i^T \\ \vdots \\ \sum_{i=1}^{\delta} \bar{B}_{\delta+i}^T \\ \vdots \\ \sum_{i=1}^{\delta} \bar{B}_{\delta(n-1)+i}^T \end{pmatrix}, \quad \hat{\bar{C}} := \begin{pmatrix} \sum_{i=1}^{\delta} \bar{C}_i^T \\ \vdots \\ \sum_{i=1}^{\delta} \bar{C}_{\delta+i}^T \\ \vdots \\ \sum_{i=1}^{\delta} \bar{C}_{\delta(n-1)+i}^T \end{pmatrix}, \quad \hat{\bar{D}} := \begin{pmatrix} \sum_{i=1}^{\delta} \bar{D}_i^T \\ \vdots \\ \sum_{i=1}^{\delta} \bar{D}_{\delta+i}^T \\ \vdots \\ \sum_{i=1}^{\delta} \bar{D}_{\delta(n-1)+i}^T \end{pmatrix}.$$

Note that we use “bar” notation to denote coarsening in variables and “hat” notation to denote coarsening in constraints. We coarsen the right-hand side f analogously and obtain the *coarse*

MILP:

$$\begin{aligned}
& \underset{v, w, x, y}{\text{minimize}} && a^T y + \bar{b}^T \bar{x} + \bar{c}^T \bar{v} + \bar{d}^T \bar{w} \\
& \text{subject to} && \hat{A}y + \hat{\bar{B}}\bar{x} + \hat{\bar{C}}\bar{v} + \hat{\bar{D}}\bar{w} \leq \hat{f} \\
& && \sum_{k=1}^K \bar{x}_{ik} \leq 1, \quad \bar{x}_{ik} \in \{0, 1\} \\
& && \sum_{k=1}^K \sum_{j=1}^{I_k} \bar{v}_{ijk} \leq 1, \quad \sum_{j=1}^{I_k} \bar{v}_{ijk} = \bar{x}_{ik}, \quad 0 \leq \bar{v}_{ijk} \leq 1 \\
& && \sum_{j=1}^J \bar{w}_{ij} \leq 1, \quad 0 \leq \bar{w}_{ij} \leq 1.
\end{aligned} \tag{2.15}$$

It follows easily that (2.15) is a relaxation of (2.13), because we have simply aggregated the constraints. We can use this fact to develop a simple algorithm that solves (2.13) by solving a sequence of tighter relaxations. The main idea is that after solving a relaxation (2.15) we can check whether all constraints in (2.13) are satisfied and add any violated constraints to (2.15). It follows easily that this algorithm is finite, because after finitely many constraints have been added to (2.15), it is equivalent to (2.13).

In practice, however, solving a sequence of MILPs (2.15) may not be efficient, because MILPs do not warm-start. Instead, we solve the LP-relaxation of the coarse model (2.15) and add violated constraints until all constraints in the LP-relaxation of the semi-coarse model (2.13) are satisfied. We use the identified constraints as the initial set of constraints for the MILP coarse model iterations. We summarize this procedure in Algorithm 1.

LP Warm-start Phase

```

Set  $l \leftarrow 0$ ;
Solve the LP-relaxation of the coarse MILP (2.15) to get solution  $(y_{\text{LP}}^0, \bar{x}_{\text{LP}}^0, \bar{v}^0, \bar{w}^0)$ ;
while  $(y_{\text{LP}}^l, \bar{x}_{\text{LP}}^l, \bar{v}^l, \bar{w}^l)$  is not feasible for the LP-relaxation of the semi-coarse MILP (2.13)
do
  Find constraints that are violated in the LP-relaxation of (2.13) and add them to (2.15);
  Solve the LP-relaxation of (2.15) with the added constraints to get
   $(y_{\text{LP}}^{l+1}, \bar{x}_{\text{LP}}^{l+1}, \bar{v}^{l+1}, \bar{w}^{l+1})$ ;
  Set  $l = l + 1$  ;

```

MILP Phase

```

Set  $k \leftarrow 0$ ;
Solve the coarse MILP (2.15) with additional constraints added in the LP Warm-start
Phase to get solution  $(y^0, \bar{x}^0, \bar{v}^0, \bar{w}^0)$ ;
while  $(y^k, \bar{x}^k, \bar{v}^k, \bar{w}^k)$  is not feasible for (2.13) do
  Find constraints that are violated in (2.13) and add them to (2.15);
  Solve the coarse MILP (2.15) with the added constraints to get  $(y^{k+1}, \bar{x}^{k+1}, \bar{v}^{k+1}, \bar{w}^{k+1})$ ;
  Set  $k = k + 1$  ;

```

Algorithm 1: Solve the semi-coarse MILP (2.13) via a sequence of coarse MILPs (2.15) using the LP-relaxation as warm-start.

Proposition 2.3. *When Algorithm 1 terminates, the solution $(y^k, \bar{x}^k, \bar{v}^k, \bar{w}^k)$ of the coarse model (2.15) with added constraints is the solution of the semi-coarse model (2.13).*

Proof. Since $(y^k, \bar{x}^k, \bar{v}^k, \bar{w}^k)$ minimizes the objective function of the semi-coarse model (2.13) and satisfies all constraints (2.13), it is the optimal solution of (2.13). \square

We show in Section 4 that our approach in Algorithm 1 is advantageous; in particular, it significantly reduces the number of MILP re-solves, as opposed to the case without LP-relaxation as warm-start. We conclude this section by providing a matrix representation of the two-level approach using Kronecker products. Recall that for two generic matrices $E \in \mathbb{R}^{n \times m}$ and $F \in \mathbb{R}^{p \times q}$, their Kronecker product is defined as

$$E \otimes F = \begin{bmatrix} e_{11}F & \cdots & e_{1m}F \\ \vdots & \ddots & \vdots \\ e_{n1}F & \cdots & e_{nm}F \end{bmatrix} \in \mathbb{R}^{np \times mq}.$$

Using (2.10), we can write

$$\bar{B} = B(I \otimes \bar{X}), \quad \bar{C} = C(I \otimes \bar{V}), \quad \bar{D} = D(I \otimes \bar{W})$$

where I is the identity matrix. Similarly, if we let $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^\delta$ be the row vector of all ones of length δ , then we can express

$$\hat{B} = (I \otimes \mathbf{1})\bar{B} = (I \otimes \mathbf{1})B(I \otimes \bar{X}), \quad \hat{C} = (I \otimes \mathbf{1})\bar{C} = (I \otimes \mathbf{1})C(I \otimes \bar{V}), \quad \hat{D} = (I \otimes \mathbf{1})\bar{D} = (I \otimes \mathbf{1})D(I \otimes \bar{W}),$$

and

$$\hat{A} = (I \otimes \mathbf{1})A, \quad \hat{f} = (I \otimes \mathbf{1})f.$$

3 Application to Cogeneration for Buildings

In this section, we apply our two-level approach to the cogeneration problem for buildings. Our MILP model (A.21) is adapted from models for cogeneration in commercial buildings (Ren and Gao, 2010; Pruitt et al., 2013). In particular, we take linearized models for fuel cells and water tank storage from the work of Pruitt et al. (2013), and we penalize on/off operations using switching cost as done by Ren and Gao (2010). While the two-level framework described in Section 2 applies to generic MILPs (1.1), the MILP model (A.21) for cogeneration entails several complex constraints as discussed in Section 3.2. As a result, additional work is required to construct appropriate semi-coarse and coarse models.

We note that our MILP model (A.21) for the cogeneration problem has the following features that are not included in existing models. First, our model has a large number of binary variables, on the order of $\mathcal{O}(10^6)$, in the second stage. This is orders of magnitude larger than the number of binary variables of Siddiqui et al. (2005), Stadler et al. (2009), Ren and Gao (2010), and Pruitt et al. (2013). This modeling feature allows considerably more degrees of freedom for on/off operation during the life time of new technologies. Second, our model contains three sets of coupling constraints that (i) couple first- and second-stage binary variables, (ii) couple second-stage variables for different technologies, and (iii) couple second-stage variables over a long time horizon (e.g., 10-20 years). The number of coupling constraints is on the same order of variables, namely, $\mathcal{O}(10^6)$. This makes the problem significantly harder because it does not lend itself to decomposition techniques such as Lagrangian relaxation.

In what follows, we describe the main characteristics of the MILP model (Section 3.1), derive the semi-coarse and coarse models (Sections 3.2 and 3.3), and discuss profile generation and selection (Section 3.4).

3.1 MILP Model

The cogeneration problem consists of two components: the investment decision and the operation planning. The investment decision concerns what new technologies to purchase, while the operation planning concerns how to dispatch units over a long-term period (e.g., 10-20 years). We formulate a two-stage MILP, where the first-stage variables model investment decisions and the second-stage variables model equipment operations. Note that both the first and second stages contain integer variables. In particular, on/off operations for new technologies in second-stage are made on an hourly basis.

Following [Siddiqui et al. \(2005\)](#); [Stadler et al. \(2009\)](#); [Ren and Gao \(2010\)](#); [Pruitt et al. \(2013\)](#), we consider five technologies: batteries, boilers, solid-oxide fuel cells (SOFCs), combined heat and power (CHP) SOFCs, and water tank storage. The selection criterion is based on the installation, operation, maintenance, fuel consumption, and carbon emission costs, while meeting electricity and heating demands. The detailed MILP model is given in [\(A.21\)](#).

3.2 Semi-Coarse Model

We begin by introducing the profiles for variables. We denote profiles by upper-case letters with the bar notation on top: $\bar{X}_{jk} \in \mathcal{P}_o$ are the on/off profiles for technology j , $\bar{P}_{jkl} \in \mathcal{P}_p$ are the production profiles associated with \bar{X}_{jk} , $\bar{U}_k \in \mathcal{P}_u$ are the profiles for power purchased from the utility, $\bar{Q}_k \in \mathcal{P}_q$ are the heat generation profiles from boilers, $\bar{B}_k, \bar{B}_k^{\text{IO}} \in \mathcal{P}_b$ are the power storage and input/output profiles for batteries, and $\bar{S}_k, \bar{S}_k^{\text{out}} \in \mathcal{P}_s$ are the heat storage and output profiles for water tanks, where $\mathcal{P}_o, \mathcal{P}_p, \mathcal{P}_u, \mathcal{P}_q, \mathcal{P}_b$, and \mathcal{P}_s are the corresponding set of profiles. We denote the h th element of a profile by $\bar{X}_{jk}(h)$ for $h \in \{1, \dots, \delta\}$. Time-dependent parameters D_t^P and D_t^Q and discount factors Y_t are concatenated into profiles \bar{D}_d^P, \bar{D}_d^Q , and $\bar{Y}_d \in \mathbb{R}^\delta$.

As described in Section 2, the variables in the original MILP model are represented by using profiles in the coarsened models. In particular, binary variables x_{ijt} are coarsened to profiles \bar{X}_{jk} , and continuous variables $p_{ijt}, u_t, b_t, s_t, s_t^{\text{out}}$, and q_t are coarsened to profiles $\bar{P}_{jkl}, \bar{U}_k, \bar{B}_k, \bar{S}_k$, and $\bar{S}_k^{\text{out}}, \bar{Q}_k \geq 0$, respectively. The coefficients for profiles are denoted by lower-case letters with a bar on top; for example, $\bar{x}_{ijdk} \in \{0, 1\}$ indicates whether profile k is selected on day d for unit i of technology j . As modeled in [\(C.22c\)](#), no more than one on/off profile can be selected for fixed i, j , and d . This, in conjunction with binary elements of \bar{X}_{jk} , results in binary-valued x_{ijt} . Similarly, non-negative profiles $\bar{P}_{jkl}, \bar{U}_k, \bar{B}_k, \bar{S}_k, \bar{S}_k^{\text{out}}$, and $\bar{Q}_k \geq 0$ and their non-negative coefficients in [\(C.22f\)](#), [\(C.22p\)](#), and [\(C.22q\)](#) result in non-negative $p_{ijt}, u_t, b_t, s_t, s_t^{\text{out}}$, and q_t .

We next turn to constraints that *do not* couple variables in time, namely, [\(A.21b\)](#), [\(A.21d\)](#), [\(A.21e\)](#), [\(A.21g\)](#), [\(A.21j\)](#), [\(A.21k\)](#), [\(A.21n\)](#), and [\(A.21o\)](#). The profile representation for these constraints is straightforward; one simply substitutes the summation of profiles as shown in [\(C.22b\)](#), [\(C.22d\)](#), [\(C.22e\)](#), [\(C.22g\)](#), [\(C.22j\)](#), [\(C.22k\)](#), [\(C.22n\)](#), and [\(C.22o\)](#), respectively. Note that all inequalities in [\(C.22\)](#) are elementwise inequalities. We do not include the production constraint [\(A.21c\)](#) in the semi-coarse model. Instead, we require that the production profile \bar{P}_{jkl} associated with the on/off profile \bar{X}_{jk} satisfy

$$R_j^{\min} \bar{X}_{jk} \leq \bar{P}_{jkl} \leq R_j^{\max} \bar{X}_{jk}. \quad (3.16)$$

This approach is justified because [\(A.21c\)](#) follows by construction due to the convex combination of coefficients for production profiles in [\(C.22f\)](#). By an analogous argument, $s_t^{\text{out}} \leq s_t$ in [\(A.21n\)](#) is a direct consequence of the following condition imposed on profiles:

$$\bar{S}_k^{\text{out}} \leq \bar{S}_k. \quad (3.17)$$

We next discuss how to coarsen variables that are coupled over time periods. The boundary conditions (A.21i) and (A.21n) can be expressed as (C.22i) and (C.22m). The maximum power purchased constraint (A.21g) can be rewritten as (C.22g), where $\mathbf{1}$ denotes the vector of all ones with length δ . We next turn to switching and storage constraints (A.21f), (A.21h), and (A.21l), whose profile representation needs additional notation. Given an on/off profile $\bar{X}_{jk} \in \{0, 1\}^\delta$, we can construct the associated switching profile

$$\bar{W}_{jk}(h) = |\bar{X}_{jk}(h+1) - \bar{X}_{jk}(h)|. \quad (3.18)$$

Now, the switching cost can be included in the objective function in (C.22). To deal with the battery storage constraint (A.21h) that couples b_t and b_t^{IO} over the horizon T , we require that the pair of profiles $(\bar{B}_k, \bar{B}_k^{\text{IO}})$ satisfy

$$\bar{B}_k(h+1) = (1 - L^P)\bar{B}_k(h) + \bar{B}_k^{\text{IO}}(h), \quad h = 1, \dots, \delta - 1. \quad (3.19)$$

We assign the same coefficient \bar{b}_{dk} to both sets of profiles \bar{B}_k and \bar{B}_k^{IO} throughout (C.22). It follows that (A.21h) is satisfied except for hours between profiles, namely, $t = d\delta$ for $d < |\mathcal{D}|$. Therefore, we introduce constraint (C.22h) to guarantee that (A.21h) holds for $t = d\delta$ for $d < |\mathcal{D}|$. We will show in Section 3.4 how to generate profiles that satisfy (3.16), (3.17), and (3.19). The heat storage constraint (A.21l) for $j = \text{chp}$ can be expressed as

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \{ [S_\delta - (1 - L^Q)I_\delta] \bar{S}_k + \bar{S}_k^{\text{out}} \} + \sum_{k \in \mathcal{P}_s} \bar{s}_{(d+1)k} E_\delta \bar{S}_k \leq (E_j^Q / E_j^P) \sum_{i, k, l} \bar{p}_{ijkl} \bar{P}_{jkl}, \quad d < |\mathcal{D}|,$$

where $I_\delta \in \mathbb{R}^{\delta \times \delta}$ is the identity matrix, $E_{\delta 1} \in \mathbb{R}^{\delta \times \delta}$ is a matrix with 1 in the $(\delta, 1)$ entry, and $S_\delta \in \mathbb{R}^{\delta \times \delta}$ is a Toeplitz matrix with only nonzero elements being 1 at the first upper-subdiagonal.

3.3 Coarse Model

We next turn to the coarse model. Note that constraints in the semi-coarse model (C.22) are elementwise equalities or inequalities involving daily profiles. We aggregate hourly constraints by summing the elements of daily profiles. Using the hat notation on the top to denote the elementwise summation of profiles, for example, $\hat{P}_{jkl} = \sum_{h=1}^{\delta} \bar{P}_{jkl}(h)$, we coarsen δ hourly constraints into a single constraint. For example, the hourly demand constraint (C.22b) is replaced by the daily demand constraint (D.23b). Similarly, the maximum hourly power demand (C.22g) is aggregated into the maximum daily demand (D.23g). The symmetric breaking constraint (D.23e) has the interpretation that the number of times unit $i + 1$ is turned on is no greater than the number of times unit i is turned on in day d . The capacity constraints for technologies (D.23d), (D.23j), (D.23n), and (D.23o) imply that the daily power/heat output is bounded by the daily capacity of the technology units purchased at the first-stage. Since (C.22h), (C.22i), and (C.22m) are scalar constraints themselves, no aggregation is applied to (D.23h), (D.23i), (D.23m) in the coarse model. Also, since the coefficients of profiles are the same for both semi-coarse and coarse models, (D.23c), (D.23f), and (D.23p) stay the same as (C.22c), (C.22f), and (C.22q) in the semi-coarse model.

3.4 Profile Generation and Selection

Recall that the on/off profiles \bar{X}_{jk} must have binary elements and the profiles \bar{P}_{jkl} , \bar{Q}_k , \bar{U}_k , \bar{B}_k , \bar{S}_k , and \bar{S}_k^{out} must have non-negative elements. In addition, we impose constraints (3.16), (3.17), and (3.19) in formulating the semi-coarse model in Section 3.2. In this section, we discuss how to generate valid profiles, and we provide suggestions for profile selections.

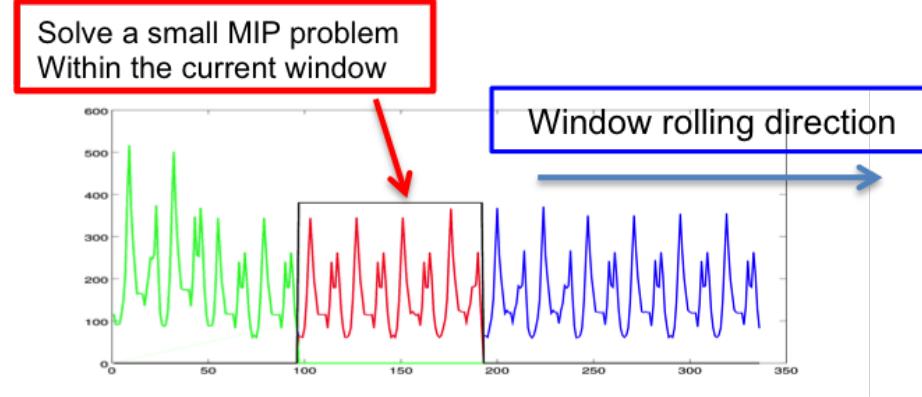


Figure 2: Illustration of the moving horizon method in Algorithm 2.

One approach to generating profiles that satisfy the above constraints is as follows. We solve a number of small instances of the original MILP (A.21), take snapshots of the second-stage solutions, and extract profiles from these snapshots. For example, consider a four-day MILP (A.21); that is, the time horizon in the second-stage problem has only four days. Solving such a four-day model, we have four snapshots of daily operation and production; in particular, we have four sets of on/off profiles for $x_{ijt} \in \{0, 1\}$ and four sets of profiles p_{ijt} , u_t , b_t , s_t , s_t^{out} , and $q_{jt} \geq 0$. We will show in Proposition 3.1 that these are valid profiles for the semi-coarse model (C.22).

The remaining question is how to choose the short-horizon MILPs. Our objective is to generate a rich set of profiles that are representative of the optimal solutions for a long-horizon MILP (A.21). To this end, we borrow the *moving-horizon approach* from model predictive control (Garcia et al., 1989; Allgöwer and Zheng, 2000). Given a long-horizon MILP, the idea is to solve MILPs over a short window, roll the window forward, and re-solve the new MILP until the window reaches the end of the horizon; see Figure 2 for an illustration. We summarize this approach and provide additional details in Algorithm 2.

Data: The parameters for MILP (A.21) with a horizon of D days, and a window of w days with $w \ll D$.

Result: Profiles $\{\bar{P}_{jkl}, \bar{Q}_k, \bar{U}_k, \bar{B}_k, \bar{S}_k, \bar{S}_k^{\text{out}}\} \in \mathbb{R}_+^\delta$, $\bar{B}_k^{\text{IO}} \in \mathbb{R}^\delta$, and $\{\bar{X}_{jk}, \bar{W}_{jk}\} \in \{0, 1\}^\delta$.

Set $k \leftarrow 0$, $\mathcal{W} = \{1, \dots, \delta w\}$, $\mathcal{R} = \{1, \dots, \delta\}$;

while $k + w \leq D$ **do**

- Solve MILP (A.21) in the current window $t \in \mathcal{W}$;
- Take snapshots of solutions $\{x_{ijt}, p_{ijt}, q_t, u_t, b_t, b_t^{\text{IO}}, s_t, s_t^{\text{out}}\}$ in $t \in \mathcal{R}$;
- Extract profiles $\{\bar{X}_{jk}, \bar{P}_{jkl}, \bar{Q}_k, \bar{U}_k, \bar{B}_k, \bar{B}_k^{\text{IO}}, \bar{S}_k, \bar{S}_k^{\text{out}}\}$, and $\bar{W}_{jk} \in \{0, 1\}^\delta$ using (3.18);
- Roll the window forward by setting $k \leftarrow k + 1$;
- if** $k + w < D$ **then** Set $\mathcal{W} = \{\delta k + 1, \dots, \delta(k + w)\}$, $\mathcal{R} = \{\delta k + 1, \dots, \delta(k + 1)\}$;
- if** $k + w = D$ **then** Set $\mathcal{W} = \mathcal{R} = \{\delta k + 1, \dots, \delta H\}$.

Algorithm 2: Moving horizon method for profile generation.

Proposition 3.1. *The moving-horizon method described in Algorithm 2 generates non-negative profiles $\{\bar{P}_{jkl}, \bar{Q}_k, \bar{U}_k, \bar{B}_k, \bar{S}_k, \bar{S}_k^{\text{out}}\} \in \mathbb{R}_+^\delta$ and binary profiles $\{\bar{X}_{jk}, \bar{W}_{jk}\} \in \{0, 1\}^\delta$. Moreover, the profile pairs $\{\bar{X}_{jk}, \bar{P}_{jkl}\}$, $\{\bar{S}_k^{\text{out}}, \bar{S}_k\}$, and $\{\bar{B}_k, \bar{B}_k^{\text{IO}}\}$ satisfy (3.16), (3.17), and (3.19), respectively.*

Proof. Non-negativity of the profiles $\{\bar{P}_{jkl}, \bar{Q}_k, \bar{U}_k, \bar{B}_k, \bar{S}_k, \bar{S}_k^{\text{out}}\}$ follows from the fact that the

second-stage solutions $\{p_{ijt}, q_t, u_t, b_t, s_t, s_t^{\text{out}}\}$ of the original MILP model (A.21) are non-negative. Similarly, $\bar{X}_{jk} \in \{0, 1\}^\delta$ follows from $x_{ijt} \in \{0, 1\}$; and \bar{W}_{jk} , constructed from (3.18), is elementwise binary. Since the production variable p_{ijt} and the on/off variable x_{ijt} satisfy (A.21c), it follows that $\{\bar{X}_{jk}, \bar{P}_{jkl}\}$ satisfies (3.16). Since the power storage b_t and power input/output b_t^{IO} satisfy (A.21h) and since the heat storage s_t and heat output s_t^{out} satisfy (A.21l), we conclude that (3.17) and (3.19) follow by construction. \square

Algorithm 2 potentially generates a huge number of profiles; hence, solving (C.22) and (D.23) with all generated profiles may be prohibitive. Instead, we select on/off profiles \bar{X}_{jk} that appear most frequently in the generated profile pool. Since the aim of the two-level approach is to reduce the problem size, it is desired that the number of on/off profiles k is much smaller than the length of profiles δ (e.g., $k \approx \delta/10$). For the production profiles \bar{P}_{jkl} associated with each \bar{X}_{jk} , we choose production profiles that have the minimum or maximum total production $\sum_{h=1}^\delta \bar{P}_{jkl}(h)$. These extreme profiles provide an envelope of other profiles; thus, their convex combination (C.22f) provides a good range of profiles for selection. Similarly, profiles with the minimum and maximum sum of absolute values are chosen for the battery storage \bar{B}_k , the heat storage \bar{S}_k , the battery input/output \bar{B}_k^{IO} , and the heat output \bar{S}_k^{out} . Further, profiles for the power purchased from the utility \bar{U}_k and the heat output from the boiler \bar{Q}_k are uniformly sampled over the period of the entire horizon.

Alternatively, we also employ a *k-means clustering algorithm* in order to cluster profiles. Given a prespecified k number of clusters, this algorithm assigns profiles to one of k clusters defined by the centroids (Jain, 2010). Since demand and pricing data for buildings tend to differ significantly in winters and summers, we set $k = 2$, and choose one profile that has the minimum least-squares distance to the centroids. In our numerical experiments in Section 4, we apply a *k*-means algorithm to the boiler output \bar{Q}_k , power purchased from the utility \bar{U}_k , heat storage output \bar{S}_k^{out} , and battery power output \bar{B}_k^{IO} profiles. The clustering approach achieves better performance in the objective function than does simple sampling heuristics for profile selection.

4 Numerical Results and Case Studies

In this section, we illustrate the effectiveness of our two-level approach using five different building examples. We demonstrate that both the semi-coarse and the coarse models allow us to find good approximate solutions in a fraction of the time compared with solving the full MILP model. The two-level approach also scales to large problems that are beyond the scope of state-of-the-art commercial MILP solvers.

4.1 Generation of Problem Instances

We use the simulation program *EnergyPlus 8.2*. (Crawley et al., 2001, 2000) to generate hourly electricity and heating demands for typical summer and winter days for five types of buildings located in Chicago, Illinois. We scale the daily demands, depending on building types, to generate weekly summer and winter demands. The scaling factors for building types are shown in Table 1. We generate yearly demand profiles by interpolating between winter and summer weeks. The demand of the i th week is given by $D_i = w_i D_{\text{wt}} + (1 - w_i) D_{\text{sm}}$, where D_{wt} and D_{sm} are the weekly demand in winter and summer, respectively, and w_i is a piecewise linear function shown in Figure 3. The resulting yearly demand profiles are perturbed by a zero mean unit variance Gaussian noise whose magnitude is 2% proportional to the magnitude of demands. We repeat this process to generate demands for multiple years.

Building Type	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.
Office	1	1	1	1	1	1/2	1/4
Supermarket	1/2	1/2	1/2	1/2	1/2	1	1
Restaurant	1/4	1/4	1/2	1/2	1	1	1/2
Hospital	1	1	1	1	1	1	1
Retail	1/4	1/2	1/2	1	1	1/4	1/4

Table 1: Scaling factors of daily demands in a week for five buildings. For example, the demand of the office building on Saturdays is scaled by half of the demand on weekdays.

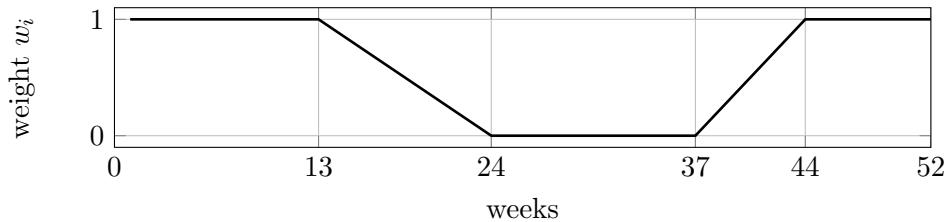


Figure 3: Piecewise linear interpolation between winter and summer weeks. The demand of the i th week is given by $D_i = w_i D_{\text{wt}} + (1 - w_i) D_{\text{sm}}$, where D_{wt} is the weekly demand in winter and D_{sm} is the weekly demand in summer.

Figure 4 shows the electricity demand in summer and winter weeks for five different buildings: an office building, a supermarket, a full-service restaurant, a hospital, and a stand-alone retail store. In addition to different peak demand patterns, the number of kilowatt-hours varies by orders of magnitude for different buildings, from tens of kilowatt-hours for the restaurant to more than a thousand kilowatt-hours for the hospital. The diversity of peak demand patterns and the wide range of magnitudes show the richness of the example set. As a result, the difficulty of the MILP model (A.21) varies significantly for different buildings; see Section 4.3.

We follow the pricing structure of [Stadler et al. \(2009\)](#). The price for electricity and gas is \$0.12 per kWh and \$0.049 per kWh, respectively. The peak demand charge is \$14.2 per kW for summer months (from June to September) and \$11.36 per kW for winter months (from October to May). These prices are fixed for each year over the second-stage time horizons.

4.2 Numerical Experiment Setup

Numerical experiments are performed on a workstation with 32 GB memory and two Intel E5430 Xeon 4-core 2.66 GHz CPUs. We implement our algorithms in AMPL ([Fourer et al., 2012](#)) to take advantage of AMPL's compact modeling syntax in profile generation, storage, and management. We use CPLEX version 12.6.1.0 as the MIP solver in AMPL. We set a 3-hour time limit and 1% relative gap as the stopping criteria for CPLEX.

4.3 Solutions of the Full Model with Short Second-Stage Horizons

Table 2 shows the problem size of the full MILP model (A.21) as a function of the number of days in the second stage. The number of binary variables, continuous variables, and constraints grows linearly with the number of days in the second-stage problem. For the one-year model, there are 1.05×10^5 binary variables, 2.71×10^5 continuous variables, and 6.11×10^5 constraints.

Table 3 shows the solutions of small problems for the restaurant example using CPLEX. We

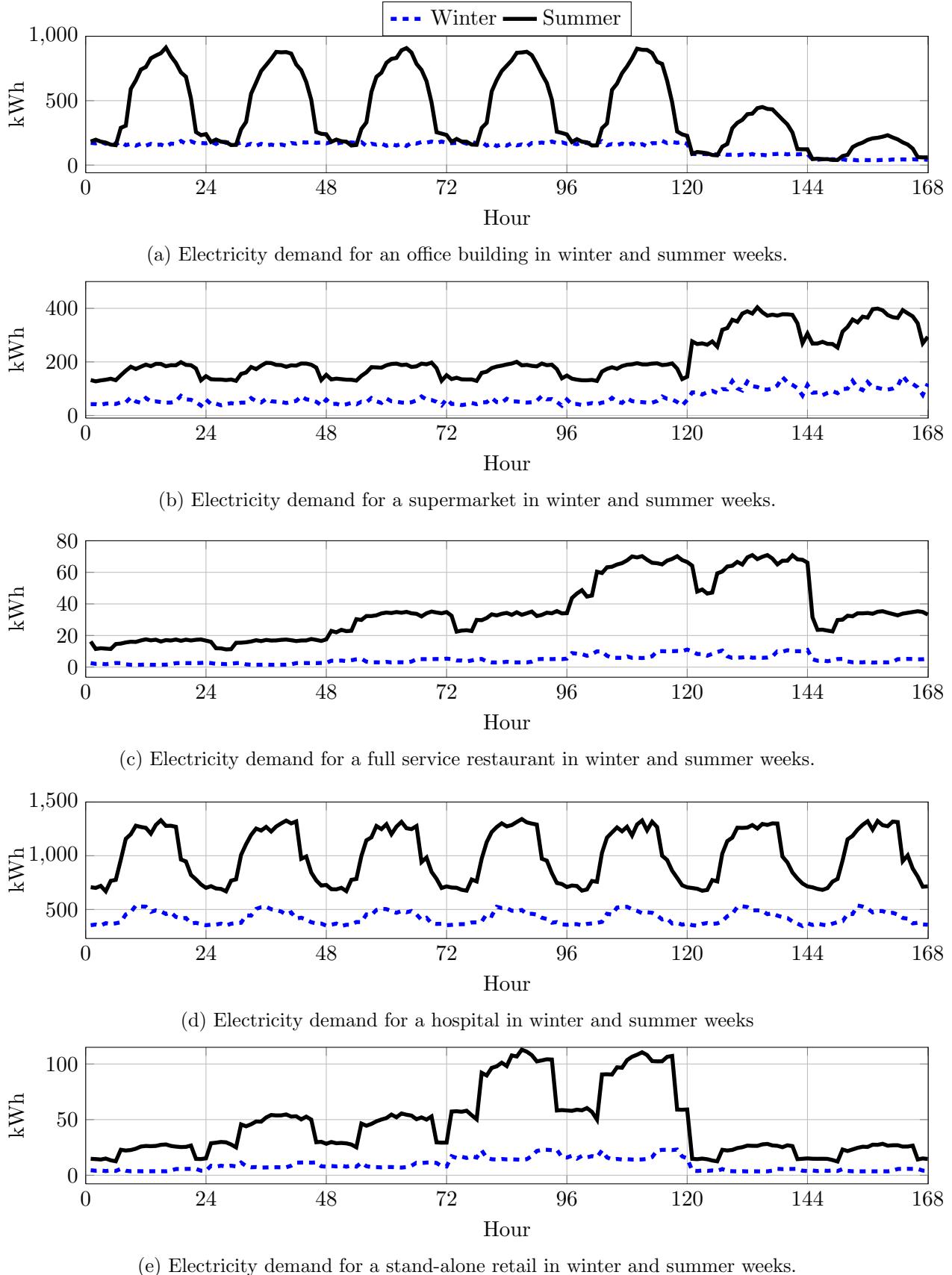


Figure 4: Electricity demand for five buildings in winter and summer weeks.

Days	Binary	Continuous	Constraint
4	1,152	2,994	6,698
7	2,016	5,226	11,738
14	4,032	10,434	23,498
28	8,064	20,850	47,018
84	24,192	62,514	$1.41 \cdot 10^5$
364	$1.05 \cdot 10^5$	$2.71 \cdot 10^5$	$6.11 \cdot 10^5$

Table 2: Problem size of the MILP model (A.21). The number of binary variables, continuous variables, and constraints grows linearly with the number of days in the second stage.

Days	Time(s)	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor
4	2	0	4.00e+03	1	1	2	0	6
7	241	2807	4.49e+05	2	1	0	1	0
14	1.21e+03	10352	2.73e+06	2	1	0	1	0
28	5.69e+03	47770	6.02e+06	2	1	0	1	0
84 [†]	1.06e+04	487	1.31e+06	6	1	0	1	0

Table 3: Solutions of the MILP model (A.21) for the restaurant example using CPLEX. The time in seconds, the number of nodes, and the number of simplex iterations grow exponentially with the horizon length in the second stage. The first-stage solutions for batteries (Bat), boilers (Boil), CHP-SOFC (Chp), Power SOFC (Pow), and water tank storage (Stor) vary with the problem size.

[†]The 84-day model reaches the 3-hour limit, but the relative MIP gap is greater than 18%.

see the exponential increase of the amount of time, the number of nodes in the branch-and-bound trees, and the number of simplex iterations with the number of days. Solving the 84-day example is beyond the capabilities of CPLEX on the designated workstation. After reaching the 3-hour time limit, the relative MIP gap for the 84-day model is still greater than 18%.¹ Our experience indicates that it is unlikely that we can solve our MILP over a ten-year time horizon.

Similar observations can be made for other building examples. In Figure 5, we see that for all five buildings, both the number of simplex iterations and the amount of solution time increase exponentially with the number of days in the second stage.

Additional numerical results are summarized in Table 6 in Appendix B. We point out that different buildings show different levels of difficulty for the MILP model (A.21). As we see in Figure 5, the five buildings differ by orders of magnitude in solution time and the number of simplex iterations. Similarly, the number of nodes in the branch-and-bound trees varies significantly over different buildings. For example, the 84-day model requires 10,384 nodes for the retail example as opposed to 200 nodes for the supermarket example; see Table 6 in Appendix B.

4.4 Solutions of Semi-Coarse Model with Long Second-Stage Horizons

In Figure 6, we compare the problem size for the full model (A.21), the semi-coarse model (C.22), and the coarse model (D.23) over a 10-year second-stage horizon. We see a roughly 8 times reduction in the number of binary variables and 12 times reduction in the number of continuous variables. The reason is that for each day with 24 hours, we pick the three most frequent on/off profiles

¹Removing the time limit does not help because then the branch-and-bound tree generated by CPLEX will consume all allowable disk space of 100 GB on the workstation.

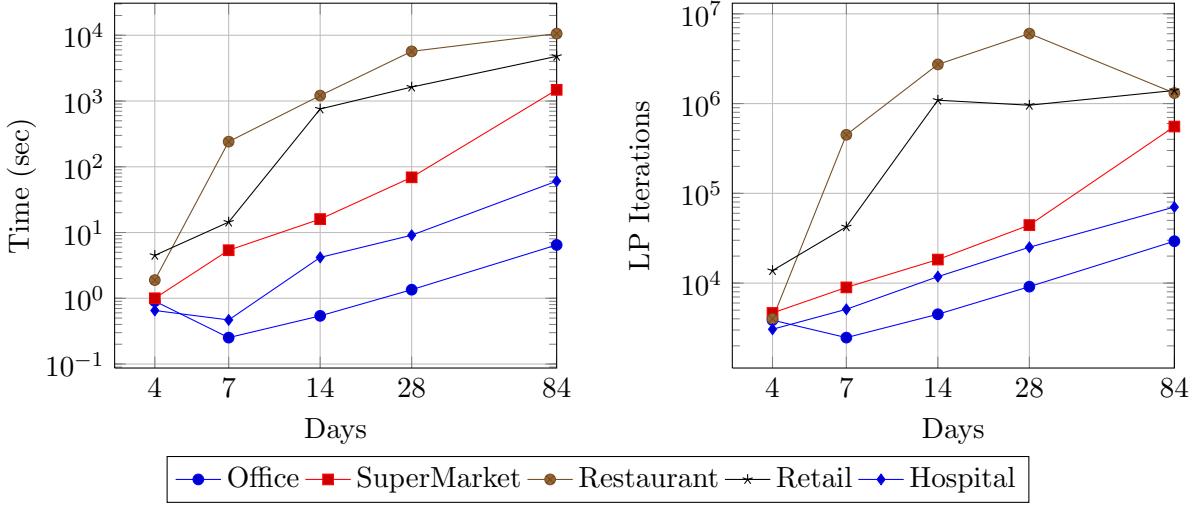


Figure 5: Exponential increase of the amount of time and the number of simplex iterations with the number of days in the second stage of the full MILP model (A.21). The number of simplex iterations drops for the 84-day *restaurant* example because CPLEX reaches 3-hour time limit.

for binary variables and two cluster centers from the k -means clustering algorithm for continuous variables. While increasing the number of profiles improves the quality of coarsened models (C.22) and (D.23), the resulting computational effort increases significantly. In practice, we find that the two-level approach strikes a good balance between solution quality and computational time with a small number of profiles (typically 2-3). Note that the large reduction in the number of constraints from the full model to the semi-coarse model. This is mainly because we have embedded the min/max production constraints (A.21c) and the switching constraints (A.21f) in the profile generation; see Section 3.2. In particular, for a 10-year model with 12 fuel cell units, there is a reduction of $12 \times 87600 \times 4 \approx 4.2 \times 10^6$ constraints (that is, a 67% of reduction in the number of constraints).

Table 4 shows the solution information for the semi-coarse model over a 10-year horizon. As opposed to the exponential increase for the full model, the solution time, the number of nodes, and the number of simplex iterations for the semi-coarse model grow more slowly. Furthermore, the first-stage solutions for the semi-coarse model are no longer sensitive to the horizon length. This result is in contrast to the variability of first-stage solutions for the full model in Table 3.

Figure 7 shows that both the number of simplex iterations and the amount of solution time scale more slowly for the semi-coarse models. For example, the number of simplex iterations for 10-year semi-coarse models is on the same order as that of the 84-day full model. The numerical results for the semi-coarse model are summarized in Table 7 in Appendix B.

4.5 Solutions of the Coarse Model with Long Second-Stage Horizons

As shown in Figure 6, the semi-coarse and coarse models have the same number of binary and continuous variables, and the semi-coarse model has about twice as many constraints than does the coarse model. Therefore, one expects that both models can be solved with a similar amount of computational effort in terms of time and simplex iterations. Since we solve a sequence of coarse models, the total computational effort often exceeds that of solving the semi-coarse model. Because of the LP warm-start phase in Algorithm 1, however, we find that the first iterate of the coarse MILPs provides remarkably good solutions of the semi-coarse model.

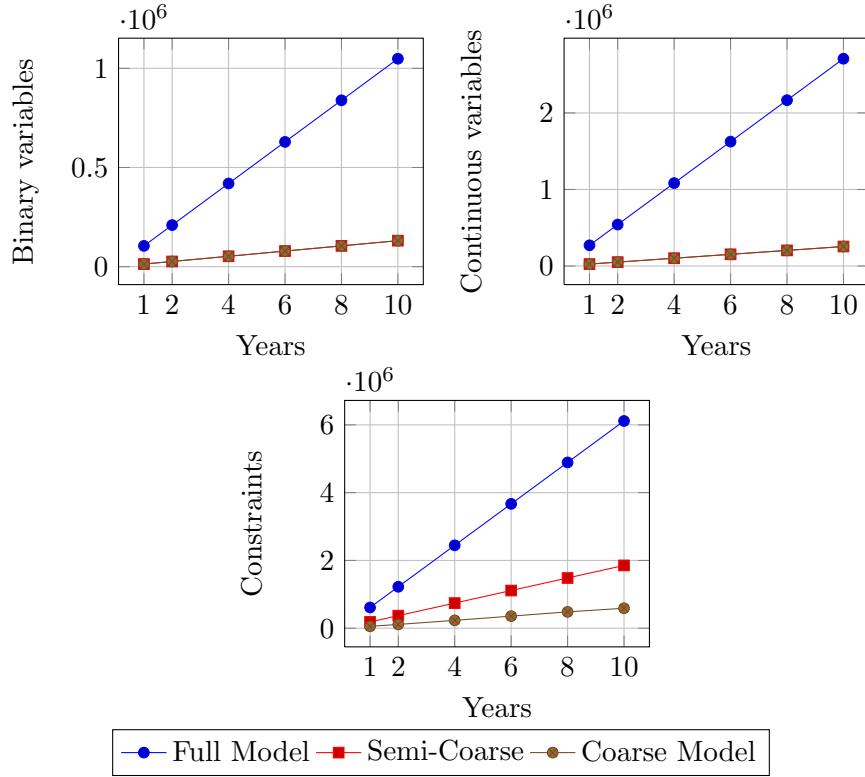


Figure 6: Problem size for the full MILP model (A.21), the semi-coarse model (C.22), and the coarse model (D.23). The number of binary and continuous variables are the same for the semi-coarse and coarse models.

Days	Time(s)	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor
364	36.4	434	21,700	0	1	1	1	1
728	157	673	57,700	0	1	1	1	1
1,456	856	1,540	$1.64 \cdot 10^5$	0	1	1	1	1
2,184	1,320	1,737	$2.48 \cdot 10^5$	0	1	1	1	1
2,912	1,520	1,529	$2.37 \cdot 10^5$	0	1	1	1	1
3,640	3,350	2,512	$3.01 \cdot 10^5$	0	1	1	1	1

Table 4: Solutions of the semi-coarse model (C.22) for the restaurant example using CPLEX. The time in seconds, the number of nodes, and the number of simplex iterations grow much more slowly than for the full model in Table 3. The first-stage solutions for batteries (Bat), boilers (Boil), CHP-SOFC (CHP), power SOFC (Pow), and water tank storage (Stor) do not change with the problem size.

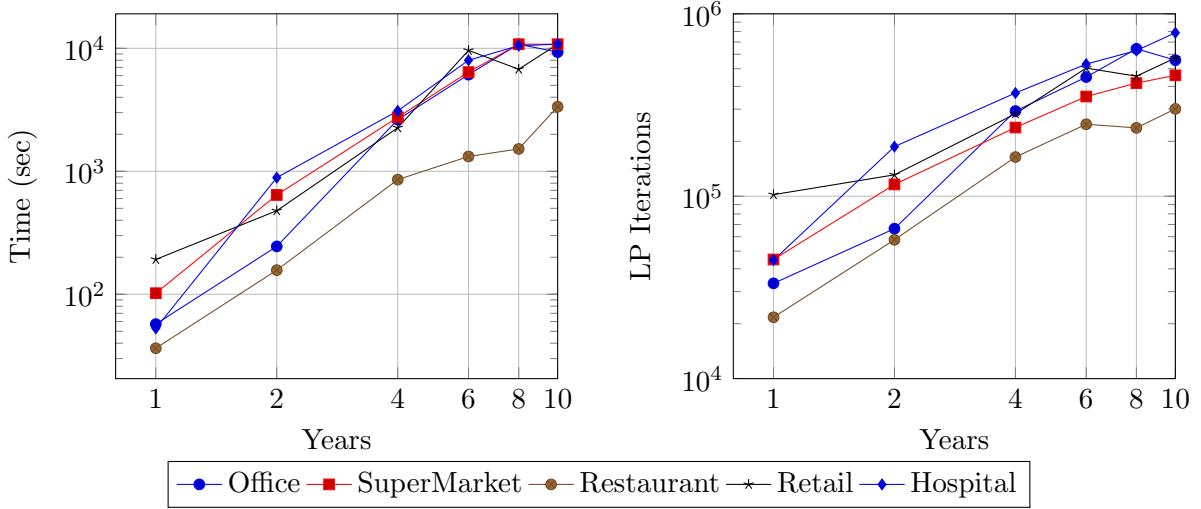


Figure 7: Solution time and number of simplex iterations for the semi-coarse model over a 10-year horizon. The growth of computational effort is considerably slower than that of the full model shown in Figure 5.

Iter	Binary	Continuous	Constraint	Objective Value
1	131,040	254,935	589,321	3,797,389.08
2	131,040	254,935	589,617	3,797,565.30
3	131,040	254,935	589,629	3,797,568.94

Iter	Time(s)	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor
1	7,821.83	3,620	$1.06 \cdot 10^6$	0	1	1	1	1
2	1,341.28	472	$2.38 \cdot 10^5$	0	1	1	1	1
3	1,125.19	938	$1.47 \cdot 10^5$	0	1	1	1	1

Table 5: Problem size of MILP iterates for a 10-year model of the restaurant example. The first iterate accounts for 76% of computational time, and the relative gap between the objective value defined in (4.20) is less than 4.7×10^{-4} .

To illustrate this point, we show in Table 5 the solution history of the MILP phase in Algorithm 1. The MILP phase converges in three coarse MILP iterates, and the first iterate requires much more computational effort than do other iterates; in particular, it accounts for 76% of total computational time. Compared with the solution from the semi-coarse model, the first iterate of the coarse MILPs provides a very good solution. The relative gap of the objective value

$$\mu = (\text{ObjVal}_{\text{semi}} - \text{ObjVal}_{\text{coar}}) / \text{ObjVal}_{\text{semi}} \quad (4.20)$$

between the semi-coarse model and the coarse model at the first iterate is less than 4.7×10^{-4} .

Figure 8 shows that the relative gap μ for all five building examples over the 10-year horizon is no greater than 0.05. This implies that the first iterate of coarse MILPs is a good approximation of the semi-coarse model. Numerical results for the first iterate of coarse MILPs are summarized in Table 8 in Appendix B.

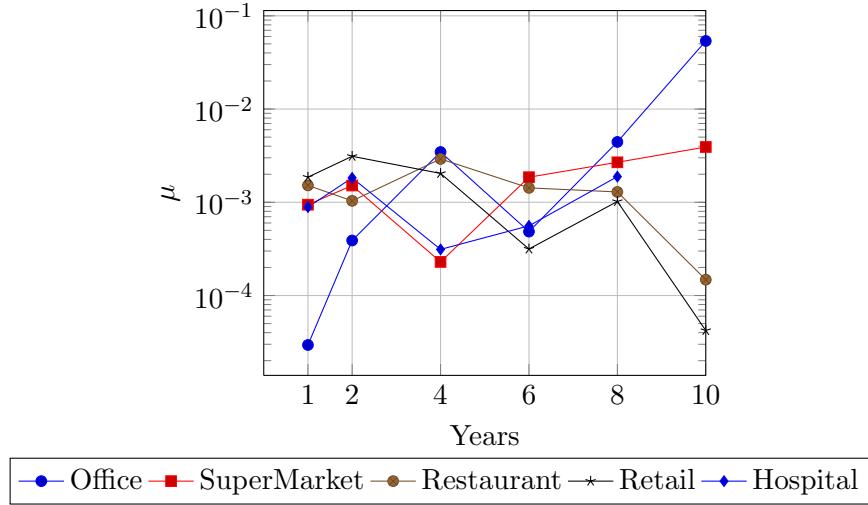


Figure 8: The relative gap μ defined in (4.20) between the first iterate of coarse MILPs and the semi-coarse model is no greater than 0.05 for all five buildings over a 10-year horizon.

4.6 Effect of the LP Warm-Start

To illustrate the effect of the LP warm-start, we consider the cogeneration problem for the restaurant example over a one-year second-stage horizon. The original MILP (A.21) has 1.05×10^5 binary variables, 2.71×10^5 continuous variables, and 6.11×10^5 constraints. Figure 9 shows the number of constraints and the computation time during the LP warm-start phase. A large number of constraints are added in the first few (3-5) iterates, resulting in more computation time at the beginning of the algorithm. As the LP warm-start phase progresses, fewer constraints are violated, and each LP-relaxation becomes easier to solve because of the warm-start. The amount of time in the LP warm-start phase is a fraction of that with one MILP re-solve in the MILP phase.

We next compare the MILP phase of Algorithm 1 with and without the LP warm-start. Figure 10 shows the number of MILP re-solves and the computation time for both cases. Without the LP warm-start, Algorithm 1 takes 6 MILP re-solves, as opposed to 2 MILP re-solves with the LP warm-start. Moreover, the MILP re-solves with the LP warm-start is up to 20 times faster than the MILPs without the LP warm-start. We note that the improvement of the LP warm-start phase is observed in all building examples.

4.7 First-Stage Solutions

Since the first-stage solutions are of primary importance, we next compare them from the full MILP, the semi-coarse, and the coarse models. In Table 6, we see that the investment decisions at the first-stage vary significantly with respect to the horizon length of the second-stage problem. For example, the number of water tanks for storage drops from six units in the 4-day model to zero units in the 7-day model, and the number of batteries increases from two units in the 28-day model to six units in the 84-day model. These numbers imply that the optimal first-stage solutions from a short-horizon problem are suboptimal solutions for a long-horizon problem. Therefore, we must solve long-horizon MILPs (A.21) as done for the semi-coarse and coarse models.

In Table 7, the first-stage solutions are much less sensitive to the horizon length at the second stage. For example, the first-stage solutions do not change over the 10-year horizon for the supermarket and restaurant examples. For the other three buildings, the ℓ_1 -norm between first-stage

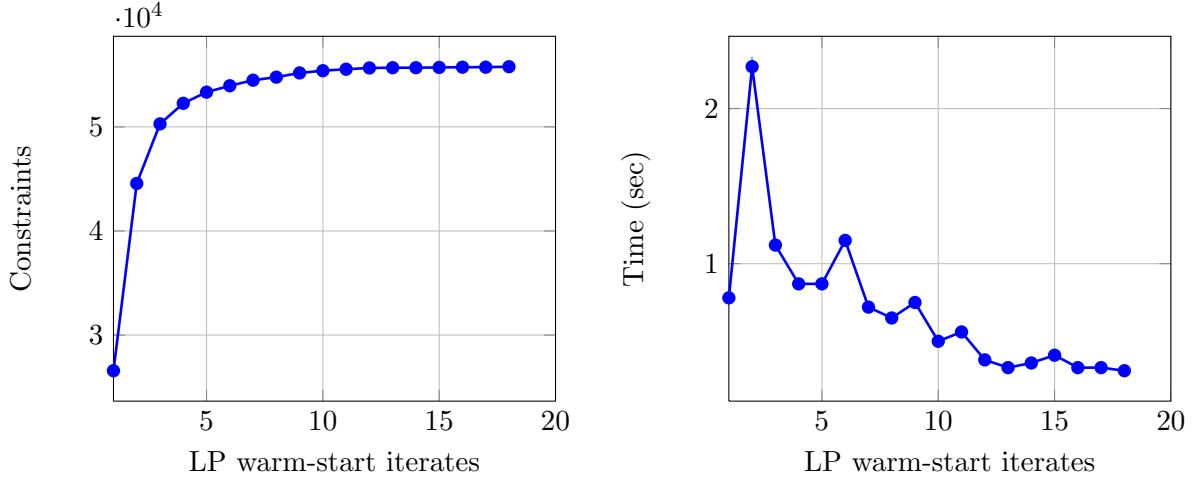


Figure 9: LP warm-start phase for the one-year restaurant example. The number of constraints (left panel) and the solution time (right panel) for LP-relaxations of (D.23).

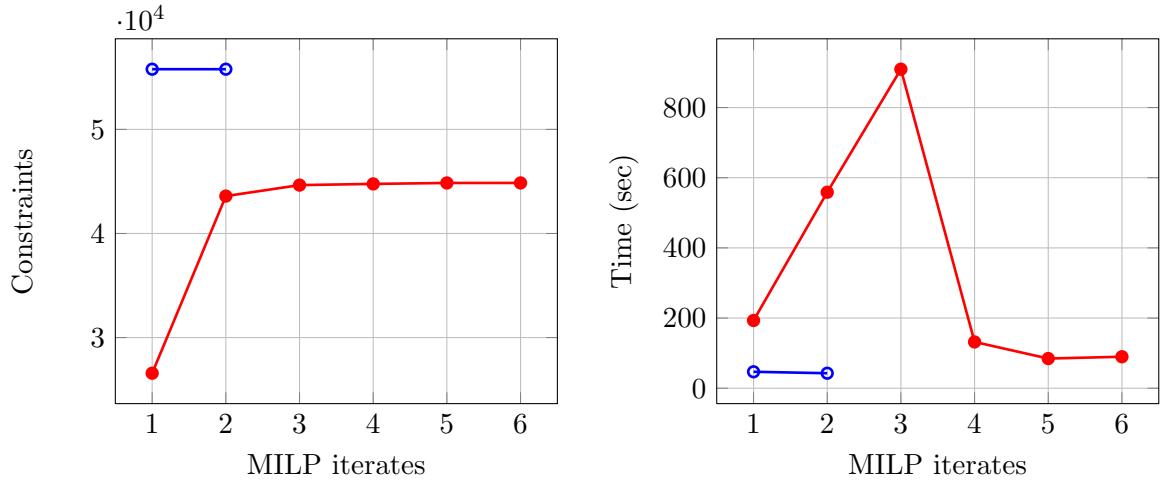


Figure 10: Effect of LP warm-start for the one-year restaurant example. Algorithm 1 with LP warm-start phase (blue line with open circles) converges with fewer MILP iterations than without LP warm-start (red line with solid circles). While more constraints are added to MILPs with LP warm-start (left panel), each MILP iterate is 5 to 20 times faster than MILPs without LP warm-start (right panel).

solutions of the same building example is no greater than 2 (i.e., at most two-unit difference of investment decisions). Similarly, the first-stage solutions at the first iterate of the coarse MILPs are insensitive to the horizon length at the second stage. The ℓ_1 -distance between the first-stage solutions of the same building example is no greater than 3.

5 Conclusions

We study two-stage MILPs with more than 1 million binary variables in the second-stage problem. We develop a two-level approach by constructing semi-coarse models (coarsened with respect to variables) and coarse models (coarsened in both variables and constraints). We show that the semi-coarse model is guaranteed to provide a feasible solution of the original MILP and hence results in an upper bound on the optimal solution. We solve a sequence of coarse MILPs with aggregated constraints that converges to the same upper bound with a finite number of steps. Furthermore, we take advantage of the LP warm-start to reduce the number of MILP re-solves.

We apply our approach to the cogeneration problem for commercial buildings. We demonstrate the effectiveness of the two-level approach using building examples with simulation data. In particular, the two-level approach allows us to obtain good approximate solutions at a fraction of the time required for solving the original MILPs. Furthermore, we show that the two-level approach scales to large problems that are beyond the capacity of state-of-the-art commercial solvers.

A number of extensions are of interest in our future work. First, how should one add new profiles in the coarsened models? Currently, we select a number of prespecified profiles and fix them throughout the MILP re-solves. Since the solution quality of coarsened models is determined by profiles, it is of interest to add new profiles dynamically that are most promising in reducing the objective value. It seems that the reduced cost associated with profiles can be obtained by solving appropriate pricing problems like those in the column generation approach (Barnhart et al., 1998). Further, it is also of interest to consider dynamic aggregations of constraints as new profiles are included. For set-partitioning constraints, a similar approach has been developed in the column generation framework (Elhallaoui et al., 2005). We plan to explore this line of research in future work.

Second, how can one obtain lower bounds on the optimal solution in the two-level framework? Since the coarse model is a relaxation of the semi-coarse model, which itself is a tightening of the full model, the coarse model is not a relaxation of the original MILP. While relaxing binary variables in the original MILP results in a lower bound, our numerical results indicate a sizable gap between this lower bound and the optimal solution. It is an open problem how to generate lower bounds using appropriate coarse MILPs.

Furthermore, advanced multilevel approaches for PDEs typically require several sweeps of fine/coarse levels in order to achieve accurate solutions (Falgout, 2006). It may be beneficial to iterate between solving coarse models to optimality and solving “partially” the fine model until a prespecified accuracy for the solution is achieved. We intend to extend our two-level framework in this direction.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract number DE-AC02-06CH11357.

A Full Model for the Cogeneration Problem

For the sake of completeness, we describe here our cogeneration model.

Index Sets. We use calligraphic upper-case letters for all sets. The set of technologies: batteries, boilers, CHP-SOFC, power SOFC, and water tank storage is denoted by \mathcal{J} . We use batt, boil, pow, chp, and stor for shortcuts. The subset $\mathcal{J}_g = \{\text{power SOFC, CHP-SOFC}\}$ denotes SOFC generation technologies that require on/off operations at the second stage. Each technology $j \in \mathcal{J}_g$ has a number of identical units, and the index set is denoted by \mathcal{U}_j . Finally, \mathcal{T} is the index set of time, \mathcal{M} is the index set of months, and $\mathcal{T}_m \subset \mathcal{T}$ is the index set of time for each month $m \in \mathcal{M}$.

Variables. We use lower-case letters to denote variables. The first-stage variables in our model are the number of units of technology $j \in \mathcal{J}$, which we denote by $y_j \in \mathbb{Z}_+$. All other variables model operation of the installed units, and they are our second-stage variables. We use the subscript t to indicate the time period: b_t and b_t^{IO} are the power storage and input/output for batteries, respectively; s_t , s_t^{in} , and s_t^{out} are the heat storage, input, and output for water tanks, respectively; p_{jt} and $q_{jt} \geq 0$ are the power and heat output of technology j , respectively; $u_t \geq 0$ is the power purchased from the utility and $u_m^{\text{max}} \geq 0$ is the maximum power purchased in month m ; $x_{ijt} \in \{0, 1\}$ indicates whether unit i of technology j operates in time period t ; and $w_{ijt} \in [0, 1]$ is an auxiliary variable to indicate whether unit i of technology j switches from t to $t + 1$. Note that we do not impose binary constraint on w_{ijt} because it takes a binary value at the solution due to the switching constraint (A.21f) and the binary constraint $x_{ijt} \in \{0, 1\}$. We use the convention $t \in \mathcal{T}$, $j \in \mathcal{J}$, $i \in \mathcal{U}_j$, $m \in \mathcal{M}$ unless explicitly mentioned.

Parameters. We use upper-case letters to denote parameters and constants: C_j is the capital and installation cost of technology j ; H is the number of hours in the lifetime of technology (e.g., $H = 87600$ for a ten-year model); T is the number of hours in the problem horizon; Y is the hourly discount factor with 3% annual interest rate (i.e., $Y = 1 - 0.03/8760$); M_j is the operation and maintenance cost of technology j ; P_t and G_t are the price for electricity and natural gas, respectively; P_m^{max} is the peak demand price for power from the utility in month m ; W_j is the cost of switching a unit of technology j on or off; S_j^c is the thermal capacity per unit for technology j ; B^c is the power capacity per unit for batteries; D_t^P and D_t^Q are the power and heat demand in period t , respectively; R_j^{min} and R_j^{max} are the minimum and maximum power output when a unit of technology j is turned on, respectively; E_j^P and E_j^Q are the power and thermal efficiency of technology j , respectively; L^P and L^Q are the average power loss in battery and heat loss in water tank storage, respectively.

Cost Function. We now formulate the complete MILP model for cogeneration problem in (A.21). The objective function consists of two parts: the capital and installation cost in the first-stage and the operation cost in the second stage. The operation cost includes the peak power usage, operation and maintenance, switching units, gas, and purchased power costs. We discount the second-stage costs with an *hourly* discount factor based on a 3% *annual* interest rate; hence $Y = 1 - 0.03/8760$. Therefore, the cost incurred in hour t is multiplied by the discount factor Y^t .² The peak demand charge at the end of month m is discounted with Y^{t_m} . To compare cost over different time horizons

²For comparison, the annual discount factor is $Y^{8760} \approx 97.044\%$, which is consistent with the 3% interest rate; and a ten-year discount factor is $Y^{87600} \approx 74.082\%$.

T , we compute the average hourly cost and multiply the result with the lifetime of technologies, H , resulting in the factor H/T in the second-stage cost in (A.21).

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in \mathcal{J}} C_j y_j + \frac{H}{T} \left\{ \sum_{m \in \mathcal{M}} Y^{t_m} P_m^{\max} u_m^{\max} \right. \\ & \left. + \sum_{t \in \mathcal{T}} Y^t \left[\sum_{j \in \mathcal{J}_g} (M_j p_{jt} + \sum_{i \in \mathcal{U}_j} W_j w_{ijt}) + G_t q_t + P_t u_t \right] \right\} \end{aligned}$$

$$\text{subject to} \quad y_j \in \mathbb{Z}, \quad 0 \leq y_j \leq |\mathcal{U}_j| \quad (\text{A.21a})$$

$$\sum_{j \in \mathcal{J}_g} p_{jt} + u_t - b_t^{\text{IO}} \geq D_t^P \quad (\text{A.21b})$$

$$R_j^{\min} x_{ijt} \leq p_{ijt} \leq R_j^{\max} x_{ijt}, \quad p_{ijt} = \sum_{i \in \mathcal{U}_j} p_{ijt} \quad j \in \mathcal{J}_g \quad (\text{A.21c})$$

$$x_{ijt} \in \{0, 1\}, \quad \sum_{i \in \mathcal{U}_j} x_{ijt} \leq y_j \quad j \in \mathcal{J}_g \quad (\text{A.21d})$$

$$x_{(i+1)jt} \leq x_{ijt} \quad j \in \mathcal{J}_g, i < |\mathcal{U}_j| \quad (\text{A.21e})$$

$$x_{ij(t+1)} - x_{ijt} \leq w_{ijt}, \quad x_{ijt} - x_{ij(t+1)} \leq w_{ijt}, \quad 0 \leq w_{ijt} \leq 1, \quad j \in \mathcal{J}_g, t < |\mathcal{T}| \quad (\text{A.21f})$$

$$0 \leq u_t \leq u_m^{\max} \quad t \in \mathcal{T}_m \quad (\text{A.21g})$$

$$b_{t+1} = (1 - L^P)b_t + b_t^{\text{IO}} \quad t < |\mathcal{T}| \quad (\text{A.21h})$$

$$b_1 = b_{|\mathcal{T}|} \quad (\text{A.21i})$$

$$0 \leq b_t \leq B^c y_{\text{batt}} \quad (\text{A.21j})$$

$$s_t^{\text{out}} + q_t \geq D_t^Q \quad (\text{A.21k})$$

$$s_{t+1} - (1 - L^Q)s_t + s_t^{\text{out}} \leq (E_{\text{chp}}^Q / E_{\text{chp}}^P)p_{\text{chp},t} \quad t < |\mathcal{T}| \quad (\text{A.21l})$$

$$s_1 = s_{|\mathcal{T}|} \quad (\text{A.21m})$$

$$s_t^{\text{out}} \leq s_t \leq S_{\text{stor}}^c y_{\text{stor}} \quad (\text{A.21n})$$

$$q_t \leq S_{\text{boil}}^c y_{\text{boil}} \quad (\text{A.21o})$$

$$b_t, p_{jt}, q_t, s_t, s_t^{\text{out}}, u_t \geq 0 \quad (\text{A.21p})$$

Constraints. The constraints in the MILP model (A.21) can be divided into three groups. The first group of constraints couples first- and second-stage variables; in particular, the number of on-units for fuel cells and the capacity of battery, storage, and boiler are bounded by the number of units purchased in the first-stage, as described in (A.21d), (A.21j), (A.21n), and (A.21o), respectively. The second group of constraints couples variables across technologies; in particular, technologies are coordinated to meet the power demand (A.21b) and the heat demand (A.21k). We note that dualizing these constraints results in decoupled variables for each technology; see, for example, Fisher (1981) and Bard (1988). However, MILP (A.21) does not lend itself to this Lagrangian relaxation approach because of coupling constraints in the third group. This group of constraints couples variables over time periods in the second stage. In particular, the battery and storage constraints (A.21h) and (A.21l) link variables over the entire horizon. In addition, constraint (A.21g), which models the maximum power purchased from the utility in each month, links

variables over time instances in each month. Moreover, constraint (A.21f), which models on/off switches of the generation units, couples binary variables x_{ijt} over two *immediate* time instances. Boundary conditions (A.21i) and (A.21m) for battery and storage couple variables at both ends of the entire horizon.

We conclude this subsection by explaining the rest of the constraints in (A.21). The number of purchased units is upper bounded in (A.21a). Since all units of the same technology are identical, we introduce a symmetry breaking constraint (A.21e) that states that if unit $i+1$ is on, then unit i must be on as well. When unit i of technology j is turned on, the power generated p_{ijt} is bounded by the minimum and maximum capacity in (A.21c). The non-negativity constraint for variables is given by (A.21p).

B Tables of Numerical Results

Type	RelGap	Days	Binary	Continuous	Constraint	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor	Time	ObjVal	MSG
1	$2.62 \cdot 10^{-4}$	4	1,152	2,994	6,698	0	3,870	0	2	5	7	8	0.9	$1.91 \cdot 10^7$	0
1	$2.84 \cdot 10^{-3}$	7	2,016	5,226	11,738	0	2,470	0	3	5	0	2	0.25	$3.01 \cdot 10^7$	0
1	$2.79 \cdot 10^{-3}$	14	4,032	10,434	23,498	0	4,500	0	3	5	0	2	0.54	$3.01 \cdot 10^7$	0
1	$4.55 \cdot 10^{-3}$	28	8,064	20,850	47,018	0	9,130	0	3	5	0	2	1.35	$3 \cdot 10^7$	0
1	$4.4 \cdot 10^{-4}$	84	24,192	62,514	$1.41 \cdot 10^5$	0	29,300	0	3	5	0	2	6.47	$2.98 \cdot 10^7$	0
1	$6.51 \cdot 10^{-3}$	364	$1.05 \cdot 10^5$	$2.71 \cdot 10^5$	$6.11 \cdot 10^5$	0	$3.88 \cdot 10^5$	0	2	5	7	3	3,950	$2.39 \cdot 10^7$	0
2	$4.27 \cdot 10^{-3}$	4	1,152	2,994	6,698	0	4,650	0	1	5	1	8	1	$1.97 \cdot 10^7$	0
2	$4.34 \cdot 10^{-6}$	7	2,016	5,226	11,738	0	8,960	0	1	0	4	0	5.38	$9.38 \cdot 10^6$	0
2	$2.89 \cdot 10^{-3}$	14	4,032	10,434	23,498	0	18,300	0	1	0	4	0	16	$9.4 \cdot 10^6$	0
2	$5.35 \cdot 10^{-3}$	28	8,064	20,850	47,018	0	44,300	0	1	0	4	0	69.1	$9.41 \cdot 10^6$	0
2	$4.49 \cdot 10^{-3}$	84	24,192	62,514	$1.41 \cdot 10^5$	200	$5.55 \cdot 10^5$	0	1	1	4	1	1,480	$9.48 \cdot 10^6$	0
2	$7.27 \cdot 10^{-2}$	364	$1.05 \cdot 10^5$	$2.71 \cdot 10^5$	$6.11 \cdot 10^5$	20	$1.75 \cdot 10^6$	0	1	5	5	5	10,800	$1.51 \cdot 10^7$	1
3	$9.93 \cdot 10^{-3}$	4	1,152	2,994	6,698	0	4,000	1	1	2	0	6	1.89	$5.01 \cdot 10^6$	0
3	$9.61 \cdot 10^{-3}$	7	2,016	5,226	11,738	2,807	$4.49 \cdot 10^5$	2	1	0	1	0	241	$1.72 \cdot 10^6$	0
3	$9.54 \cdot 10^{-3}$	14	4,032	10,434	23,498	10,352	$2.73 \cdot 10^6$	2	1	0	1	0	1,210	$1.71 \cdot 10^6$	0
3	$9.95 \cdot 10^{-3}$	28	8,064	20,850	47,018	47,770	$6.02 \cdot 10^6$	2	1	0	1	0	5,690	$1.72 \cdot 10^6$	0
3	0.18	84	24,192	62,514	$1.41 \cdot 10^5$	487	$1.31 \cdot 10^6$	6	1	0	1	0	10,600	$2.05 \cdot 10^6$	1
3	0.29	364	$1.05 \cdot 10^5$	$2.71 \cdot 10^5$	$6.11 \cdot 10^5$	0	$7.16 \cdot 10^5$	0	1	0	0	0	10,000	$4.53 \cdot 10^6$	1
4	$7.19 \cdot 10^{-3}$	4	1,152	2,994	6,698	30	13,800	1	1	2	0	8	4.49	$5.7 \cdot 10^6$	0
4	$9.77 \cdot 10^{-3}$	7	2,016	5,226	11,738	461	42,400	1	1	3	1	8	14.4	$5.98 \cdot 10^6$	0
4	$9.44 \cdot 10^{-3}$	14	4,032	10,434	23,498	10,080	$1.09 \cdot 10^6$	2	1	1	1	7	757	$4.14 \cdot 10^6$	0
4	$8.6 \cdot 10^{-3}$	28	8,064	20,850	47,018	3,490	$9.59 \cdot 10^5$	2	1	2	1	8	1,630	$4.83 \cdot 10^6$	0
4	$9.79 \cdot 10^{-3}$	84	24,192	62,514	$1.41 \cdot 10^5$	10,384	$1.4 \cdot 10^6$	2	1	1	1	1	4,780	$4.71 \cdot 10^6$	0
4	0.12	364	$1.05 \cdot 10^5$	$2.71 \cdot 10^5$	$6.11 \cdot 10^5$	0	$8.24 \cdot 10^5$	1	1	2	1	1	9,430	$5.44 \cdot 10^6$	1
5	$3.47 \cdot 10^{-7}$	4	1,152	2,994	6,698	0	3,060	6	5	5	7	8	0.65	$7.25 \cdot 10^7$	0
5	$5.27 \cdot 10^{-3}$	7	2,016	5,226	11,738	0	5,110	6	4	5	7	8	0.47	$6.83 \cdot 10^7$	0
5	$6.77 \cdot 10^{-3}$	14	4,032	10,434	23,498	0	11,800	6	4	5	7	8	4.18	$6.42 \cdot 10^7$	0
5	$9.35 \cdot 10^{-3}$	28	8,064	20,850	47,018	0	25,100	6	4	5	7	8	9.1	$6.74 \cdot 10^7$	0
5	$9.74 \cdot 10^{-3}$	84	24,192	62,514	$1.41 \cdot 10^5$	0	70,400	6	5	5	7	8	60.5	$6.52 \cdot 10^7$	0
5	$7.32 \cdot 10^{-3}$	364	$1.05 \cdot 10^5$	$2.71 \cdot 10^5$	$6.11 \cdot 10^5$	0	$3.14 \cdot 10^5$	6	5	5	7	8	638	$6.51 \cdot 10^7$	0

Table 6: Problem size and AMPL/CPLEX solution information for the cogeneration problem (A.21) for five buildings: 1-Office, 2-Supermarket, 3-Restaurant, 4-Retail, 5-Hospital. Model instances that reach 3-hour time limit are indicated by MSG=1.

Type	RelGap	Days	Binary	Continuous	Constraint	Nodes	LP-iter	Bat	Boil	Chp	Pow	Stor	Time	ObjVal	MSG
1	$9.45 \cdot 10^{-3}$	364	13,104	25,498	$1.85 \cdot 10^5$	590	33,300	0	3	5	7	2	57.2	$2.8 \cdot 10^7$	0
1	$8.53 \cdot 10^{-3}$	728	26,208	50,991	$3.71 \cdot 10^5$	660	66,400	0	3	5	7	2	245	$2.76 \cdot 10^7$	0
1	$6.71 \cdot 10^{-3}$	1,456	52,416	$1.02 \cdot 10^5$	$7.41 \cdot 10^5$	4,515	$2.93 \cdot 10^5$	0	3	4	7	2	2,630	$2.67 \cdot 10^7$	0
1	$9.66 \cdot 10^{-3}$	2,184	78,624	$1.53 \cdot 10^5$	$1.11 \cdot 10^6$	6,514	$4.5 \cdot 10^5$	0	3	4	7	2	6,110	$2.61 \cdot 10^7$	0
1	$1.31 \cdot 10^{-2}$	2,912	$1.05 \cdot 10^5$	$2.04 \cdot 10^5$	$1.48 \cdot 10^6$	514	$6.43 \cdot 10^5$	0	3	5	7	2	10,800	$2.54 \cdot 10^7$	1
1	$6.97 \cdot 10^{-2}$	3,640	$1.31 \cdot 10^5$	$2.55 \cdot 10^5$	$1.85 \cdot 10^6$	488	$5.57 \cdot 10^5$	0	3	5	7	2	9,310	$2.62 \cdot 10^7$	1
2	$9.43 \cdot 10^{-3}$	364	13,104	25,498	$1.85 \cdot 10^5$	1,234	45,000	0	3	5	4	2	102	$1.84 \cdot 10^7$	0
2	$9.98 \cdot 10^{-3}$	728	26,208	50,991	$3.71 \cdot 10^5$	4,130	$1.16 \cdot 10^5$	0	3	5	4	2	642	$1.8 \cdot 10^7$	0
2	$9.58 \cdot 10^{-3}$	1,456	52,416	$1.02 \cdot 10^5$	$7.41 \cdot 10^5$	7,213	$2.38 \cdot 10^5$	0	3	5	4	2	2,750	$1.75 \cdot 10^7$	0
2	$6.87 \cdot 10^{-3}$	2,184	78,624	$1.53 \cdot 10^5$	$1.11 \cdot 10^6$	10,267	$3.52 \cdot 10^5$	0	3	5	4	2	6,410	$1.7 \cdot 10^7$	0
2	$1.15 \cdot 10^{-2}$	2,912	$1.05 \cdot 10^5$	$2.04 \cdot 10^5$	$1.48 \cdot 10^6$	9,966	$4.17 \cdot 10^5$	0	3	5	4	2	10,800	$1.66 \cdot 10^7$	1
2	$1.83 \cdot 10^{-2}$	3,640	$1.31 \cdot 10^5$	$2.55 \cdot 10^5$	$1.85 \cdot 10^6$	5,479	$4.6 \cdot 10^5$	0	3	5	4	2	10,800	$1.62 \cdot 10^7$	1
3	$9.58 \cdot 10^{-3}$	364	13,104	25,498	$1.85 \cdot 10^5$	434	21,700	0	1	1	1	1	36.4	$4.32 \cdot 10^6$	0
3	$8.29 \cdot 10^{-3}$	728	26,208	50,991	$3.71 \cdot 10^5$	673	57,700	0	1	1	1	1	157	$4.27 \cdot 10^6$	0
3	$6.97 \cdot 10^{-3}$	1,456	52,416	$1.02 \cdot 10^5$	$7.41 \cdot 10^5$	1,540	$1.64 \cdot 10^5$	0	1	1	1	1	856	$4.14 \cdot 10^6$	0
3	$5.08 \cdot 10^{-3}$	2,184	78,624	$1.53 \cdot 10^5$	$1.11 \cdot 10^6$	1,737	$2.48 \cdot 10^5$	0	1	1	1	1	1,320	$4.02 \cdot 10^6$	0
3	$4.63 \cdot 10^{-3}$	2,912	$1.05 \cdot 10^5$	$2.04 \cdot 10^5$	$1.48 \cdot 10^6$	1,529	$2.37 \cdot 10^5$	0	1	1	1	1	1,520	$3.91 \cdot 10^6$	0
3	$4.33 \cdot 10^{-3}$	3,640	$1.31 \cdot 10^5$	$2.55 \cdot 10^5$	$1.85 \cdot 10^6$	2,512	$3.01 \cdot 10^5$	0	1	1	1	1	3,350	$3.8 \cdot 10^6$	0
4	$9.58 \cdot 10^{-3}$	364	13,104	22,950	$1.85 \cdot 10^5$	3,500	$1.02 \cdot 10^5$	0	1	2	1	1	192	$6.12 \cdot 10^6$	0
4	$7.56 \cdot 10^{-3}$	728	26,208	45,895	$3.71 \cdot 10^5$	4,085	$1.31 \cdot 10^5$	0	1	2	1	1	478	$6.03 \cdot 10^6$	0
4	$9.88 \cdot 10^{-3}$	1,456	52,416	91,785	$7.41 \cdot 10^5$	9,555	$2.83 \cdot 10^5$	0	1	3	1	1	2,260	$5.87 \cdot 10^6$	0
4	$1.08 \cdot 10^{-2}$	2,184	78,624	$1.38 \cdot 10^5$	$1.11 \cdot 10^6$	27,269	$5.04 \cdot 10^5$	0	1	1	1	1	9,620	$5.71 \cdot 10^6$	1
4	$9.75 \cdot 10^{-3}$	2,912	$1.05 \cdot 10^5$	$1.84 \cdot 10^5$	$1.48 \cdot 10^6$	11,259	$4.55 \cdot 10^5$	0	1	2	1	1	6,760	$5.54 \cdot 10^6$	0
4	$1.08 \cdot 10^{-2}$	3,640	$1.31 \cdot 10^5$	$2.29 \cdot 10^5$	$1.85 \cdot 10^6$	10,494	$5.73 \cdot 10^5$	0	1	1	1	1	10,800	$5.4 \cdot 10^6$	1
5	$9.94 \cdot 10^{-3}$	364	13,104	27,318	$1.85 \cdot 10^5$	506	44,600	4	7	5	7	2	52.8	$7.26 \cdot 10^7$	0
5	$7.2 \cdot 10^{-3}$	728	26,208	54,631	$3.71 \cdot 10^5$	4,250	$1.87 \cdot 10^5$	4	6	5	7	2	888	$7.13 \cdot 10^7$	0
5	$9.6 \cdot 10^{-3}$	1,456	52,416	$1.09 \cdot 10^5$	$7.41 \cdot 10^5$	7,180	$3.68 \cdot 10^5$	3	6	5	7	2	3,100	$6.95 \cdot 10^7$	0
5	$9.37 \cdot 10^{-3}$	2,184	78,624	$1.64 \cdot 10^5$	$1.11 \cdot 10^6$	10,324	$5.3 \cdot 10^5$	4	6	5	7	2	8,000	$6.74 \cdot 10^7$	0
5	$1.55 \cdot 10^{-2}$	2,912	$1.05 \cdot 10^5$	$2.19 \cdot 10^5$	$1.48 \cdot 10^6$	5,686	$6.29 \cdot 10^5$	4	6	5	7	2	10,500	$6.59 \cdot 10^7$	1
5	$3.64 \cdot 10^{-2}$	3,640	$1.31 \cdot 10^5$	$2.73 \cdot 10^5$	$1.85 \cdot 10^6$	2,966	$7.87 \cdot 10^5$	3	6	5	7	2	10,800	$6.55 \cdot 10^7$	1

Table 7: Problem size and AMPL/CPLEX solution information for the semi-coarse model (C.22) for five buildings: 1-Office, 2-Supermarket, 3-Restaurant, 4-Retail, 5-Hospital. Model instances that reach 3-hour time limit are indicated by MSG=1.

Building	Year	Binary	Continuous	Constraint	Nodes	LP-iter	Time	Bat	Boil	Chp	Pow	Stor	ObjVal	MSG
1	1	13,104	25,498	67,251	550	34,876	35.09	0	3	5	7	2	$2.8 \cdot 10^7$	0
1	2	26,208	50,991	$1.34 \cdot 10^5$	2,507	$1.53 \cdot 10^5$	324.85	0	3	5	7	2	$2.76 \cdot 10^7$	0
1	4	52,416	$1.02 \cdot 10^5$	$2.73 \cdot 10^5$	504	$1.29 \cdot 10^5$	415.87	0	3	5	7	2	$2.68 \cdot 10^7$	0
1	6	78,624	$1.53 \cdot 10^5$	$4.13 \cdot 10^5$	6,166	$4.52 \cdot 10^5$	3,104.26	0	3	5	7	2	$2.61 \cdot 10^7$	0
1	8	$1.05 \cdot 10^5$	$2.04 \cdot 10^5$	$5.36 \cdot 10^5$	8,482	$6.13 \cdot 10^5$	6,312.24	0	3	5	7	2	$2.53 \cdot 10^7$	0
1	10	$1.31 \cdot 10^5$	$2.55 \cdot 10^5$	$6.85 \cdot 10^5$	7,864	$7.42 \cdot 10^5$	10,453.39	0	3	5	7	2	$2.48 \cdot 10^7$	1
2	1	13,104	25,498	68,345	1,410	49,815	96.47	0	3	5	4	2	$1.84 \cdot 10^7$	0
2	2	26,208	50,991	$1.4 \cdot 10^5$	3,774	$1.21 \cdot 10^5$	464.67	0	3	5	5	2	$1.8 \cdot 10^7$	0
2	4	52,416	$1.02 \cdot 10^5$	$2.95 \cdot 10^5$	11,443	$3.05 \cdot 10^5$	3,019.47	0	3	5	4	2	$1.75 \cdot 10^7$	0
2	6	78,624	$1.53 \cdot 10^5$	$4.37 \cdot 10^5$	10,343	$3.63 \cdot 10^5$	3,900.92	0	3	5	4	2	$1.7 \cdot 10^7$	0
2	8	$1.05 \cdot 10^5$	$2.04 \cdot 10^5$	$5.87 \cdot 10^5$	9,474	$4.39 \cdot 10^5$	7,293.15	0	3	5	4	2	$1.66 \cdot 10^7$	0
2	10	$1.31 \cdot 10^5$	$2.55 \cdot 10^5$	$7.37 \cdot 10^5$	11,718	$5.6 \cdot 10^5$	10,683.5	0	3	5	7	2	$1.62 \cdot 10^7$	1
3	1	13,104	25,498	55,782	520	38,780	46.85	0	1	1	1	1	$4.33 \cdot 10^6$	0
3	2	26,208	50,991	$1.12 \cdot 10^5$	471	64,663	106.27	0	1	1	1	1	$4.26 \cdot 10^6$	0
3	4	52,416	$1.02 \cdot 10^5$	$2.32 \cdot 10^5$	3,474	$2.2 \cdot 10^5$	1,102.5	0	1	1	1	1	$4.13 \cdot 10^6$	0
3	6	78,624	$1.53 \cdot 10^5$	$3.56 \cdot 10^5$	2,519	$2.06 \cdot 10^5$	1,215.44	0	1	1	1	1	$4.03 \cdot 10^6$	0
3	8	$1.05 \cdot 10^5$	$2.04 \cdot 10^5$	$4.82 \cdot 10^5$	10,529	$6.58 \cdot 10^5$	7,214.81	0	1	1	1	1	$3.9 \cdot 10^6$	0
3	10	$1.31 \cdot 10^5$	$2.55 \cdot 10^5$	$5.89 \cdot 10^5$	3,620	$1.06 \cdot 10^6$	7,821.83	0	1	1	1	1	$3.8 \cdot 10^6$	0
4	1	13,104	22,950	56,465	1,600	$1.66 \cdot 10^5$	256.57	0	1	2	1	1	$6.14 \cdot 10^6$	0
4	2	26,208	45,895	$1.15 \cdot 10^5$	2,508	$4.11 \cdot 10^5$	721.43	0	1	2	1	1	$6.05 \cdot 10^6$	0
4	4	52,416	91,785	$2.29 \cdot 10^5$	38,110	$2.39 \cdot 10^6$	7,938.05	0	1	2	1	1	$5.88 \cdot 10^6$	0
4	6	78,624	$1.38 \cdot 10^5$	$3.37 \cdot 10^5$	45,080	$1.15 \cdot 10^6$	10,705.24	0	1	1	1	1	$5.71 \cdot 10^6$	1
4	8	$1.05 \cdot 10^5$	$1.84 \cdot 10^5$	$4.61 \cdot 10^5$	25,554	$6.42 \cdot 10^5$	9,382.93	0	1	1	1	1	$5.55 \cdot 10^6$	1
4	10	$1.31 \cdot 10^5$	$2.29 \cdot 10^5$	$5.81 \cdot 10^5$	15,413	$6.33 \cdot 10^5$	9,669.02	0	1	1	1	1	$5.4 \cdot 10^6$	1
5	1	13,104	27,318	68,979	1,759	90,264	178.67	4	7	5	7	2	$7.26 \cdot 10^7$	0
5	2	26,208	54,631	$1.39 \cdot 10^5$	3,169	$1.73 \cdot 10^5$	761.49	4	6	5	7	2	$7.14 \cdot 10^7$	0
5	4	52,416	$1.09 \cdot 10^5$	$2.78 \cdot 10^5$	8,328	$3.76 \cdot 10^5$	3,633.01	3	6	5	7	2	$6.95 \cdot 10^7$	0
5	6	78,624	$1.64 \cdot 10^5$	$4.16 \cdot 10^5$	10,835	$5.36 \cdot 10^5$	6,104.61	4	6	5	7	2	$6.75 \cdot 10^7$	0
5	8	$1.05 \cdot 10^5$	$2.19 \cdot 10^5$	$5.55 \cdot 10^5$	10,707	$7.04 \cdot 10^5$	10,826.62	4	6	5	7	2	$6.58 \cdot 10^7$	1
5	10	$1.31 \cdot 10^5$	$2.73 \cdot 10^5$	$6.91 \cdot 10^5$	7,376	$7.96 \cdot 10^5$	10,227.72	4	6	5	7	2	$6.42 \cdot 10^7$	1

Table 8: Problem size and AMPL/CPLEX solution information for the first iteration of the coarse model (C.22) for five buildings: 1-Office, 2-Supermarket, 3-Restaurant, 4-Retail, 5-Hospital. Model instances that reach 3-hour time limit are indicated by MSG=1.

C Semi-Coarse Model for the Cogeneration Problem

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in \mathcal{J}} C_j y_j + \frac{H}{T} \left\{ \sum_{m \in \mathcal{M}} Y^{t_m} P_m^{\max} u_m^{\max} + \sum_d \bar{Y}_d^T \left[\sum_{j,i,k,l} \bar{p}_{ijkl} M_j \bar{P}_{jkl} \right. \right. \\ & \left. \left. + \sum_{d,j,i,k} \bar{x}_{ijdk} W_j \bar{W}_{jk} + \sum_{d,k \in \mathcal{P}_q} \bar{q}_{dk} \bar{Q}_k + \sum_{d,k \in \mathcal{P}_u} \bar{u}_{dk} \bar{U}_k \right] \right\} \end{aligned}$$

$$\text{subject to} \quad y_j \in \mathbb{Z}, \quad 0 \leq y_j \leq |\mathcal{U}_j| \quad j \in \mathcal{J} \quad (\text{C.22a})$$

$$\sum_{j,i,k,l} \bar{p}_{ijkl} \bar{P}_{jkl} + \sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \bar{U}_k - \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \bar{B}_k^{\text{IO}} \geq \bar{D}_d^P \quad (\text{C.22b})$$

$$\bar{x}_{ijdk} \in \{0, 1\}, \quad \sum_{k \in \mathcal{P}_o} \bar{x}_{ijdk} \leq 1 \quad (\text{C.22c})$$

$$\sum_{i,k} \bar{x}_{ijdk} \bar{X}_{jk} \leq y_j \mathbf{1} \quad (\text{C.22d})$$

$$\sum_{k \in \mathcal{P}_o} \bar{x}_{(i+1)j dk} \bar{X}_{jk} \leq \sum_{k \in \mathcal{P}_o} \bar{x}_{ijdk} \bar{X}_{jk} \quad i < |\mathcal{U}_j| \quad (\text{C.22e})$$

$$\bar{p}_{ijdk} \in [0, 1], \quad \sum_{l \in \mathcal{P}_p} \bar{p}_{ijkl} = \bar{x}_{ijdk}, \quad \sum_{k \in \mathcal{P}_o} \sum_{l \in \mathcal{P}_p} \bar{p}_{ijkl} \leq 1 \quad (\text{C.22f})$$

$$0 \leq \sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \bar{U}_k \leq u_m^{\max} \mathbf{1} \quad d \in \mathcal{D}_m \quad (\text{C.22g})$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{(d+1)k} \bar{B}_k(1) = \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} [(1 - L^P) \bar{B}_k(\delta) + \bar{B}_k^{\text{IO}}(\delta)] \quad d < |\mathcal{D}| \quad (\text{C.22h})$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{1k} \bar{B}_k(1) = \sum_{k \in \mathcal{P}_b} \bar{b}_{|\mathcal{D}|k} \bar{B}_k(\delta) \quad (\text{C.22i})$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \bar{B}_k \leq B^c y_{\text{batt}} \mathbf{1} \quad (\text{C.22j})$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \bar{S}_k^{\text{out}} + \sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \bar{Q}_k \geq \bar{D}_d^Q \quad (\text{C.22k})$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} [S_\delta - (1 - L^Q) I_\delta] \bar{S}_k + \sum_{k \in \mathcal{P}_s} \bar{s}_{(d+1)k} E_\delta \bar{S}_k \quad d < |\mathcal{D}|$$

$$+ \sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \bar{S}_k^{\text{out}} \leq (E_j^Q / E_j^P) \sum_{i,k,l} \bar{p}_{ijkl} \bar{P}_{jkl} \quad j = \text{chp} \quad (\text{C.22l})$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{1k} \bar{S}_k(1) = \sum_{k \in \mathcal{P}_s} \bar{s}_{|\mathcal{D}|k} \bar{S}_k(\delta) \quad (\text{C.22m})$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \bar{S}_k^{\text{out}} \leq S_{\text{stor}}^c y_{\text{stor}} \mathbf{1} \quad (\text{C.22n})$$

$$\sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \bar{Q}_k \leq S_{\text{boil}}^c y_{\text{boil}} \mathbf{1} \quad (\text{C.22o})$$

$$\bar{u}_{dk}, \quad \bar{b}_{dk}, \quad \bar{s}_{dk}, \quad \bar{q}_{dk} \in [0, 1] \quad (\text{C.22p})$$

$$\sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \leq 1, \quad \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \leq 1, \quad \sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \leq 1, \quad \sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \leq 1. \quad (\text{C.22q})$$

D Coarse Model for the Cogeneration Problem

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in \mathcal{J}} C_j y_j + \frac{H}{T} \left\{ \sum_{m \in \mathcal{M}} Y^{t_m} P_m^{\max} u_m^{\max} + \sum_d \bar{Y}_d^T \left[\sum_{j,i,k,l} \bar{p}_{ijkl} M_j \bar{P}_{jkl} \right. \right. \\ & \left. \left. + \sum_{d,j,i,k} \bar{x}_{ijkl} W_j \bar{W}_{jk} + \sum_{d,k \in \mathcal{P}_u} \bar{u}_{dk} \bar{U}_k + \sum_{d,k \in \mathcal{P}_q} \bar{q}_{dk} \bar{Q}_k \right] \right\} \end{aligned}$$

$$\text{subject to} \quad y_j \in \mathbb{Z}, \quad 0 \leq y_j \leq |\mathcal{U}_j| \quad j \in \mathcal{J} \quad (\text{D.23a})$$

$$\sum_{j,i,k,l} \bar{p}_{ijkl} \hat{\bar{P}}_{jkl} + \sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \hat{\bar{U}}_k - \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \hat{\bar{B}}_k^{\text{IO}} \geq \hat{\bar{D}}_d^P \quad (\text{D.23b})$$

$$\bar{x}_{ijkl} \in \{0, 1\}, \quad \sum_{k \in \mathcal{P}_o} \bar{x}_{ijkl} \leq 1 \quad (\text{D.23c})$$

$$\sum_{i,k} \bar{x}_{ijkl} \hat{\bar{X}}_{jk} \leq \delta \cdot y_j \quad (\text{D.23d})$$

$$\sum_{k \in \mathcal{P}_o} \bar{x}_{ijkl} \hat{\bar{X}}_{jk} \geq \sum_{k \in \mathcal{P}_o} \bar{x}_{(i+1)jkl} \hat{\bar{X}}_{jk} \quad i < |\mathcal{U}_j| \quad (\text{D.23e})$$

$$\bar{p}_{ijkl} \in [0, 1], \quad \sum_{l \in \mathcal{P}_p} \bar{p}_{ijkl} = \bar{x}_{ijkl}, \quad \sum_{k \in \mathcal{P}_o} \sum_{l \in \mathcal{P}_p} \bar{p}_{ijkl} \leq 1 \quad (\text{D.23f})$$

$$0 \leq \sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \hat{\bar{U}}_k \leq \delta \cdot u_m^{\max} \quad d \in \mathcal{D}_m \quad (\text{D.23g})$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{(d+1)k} \bar{B}_k(1) = \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} ((1 - L^P) \bar{B}_k(\delta) + \bar{B}_k^{\text{IO}}(\delta)) \quad d < |\mathcal{D}| \quad (\text{D.23h})$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{1k} \bar{B}_k(1) = \sum_{k \in \mathcal{P}_b} \bar{b}_{|\mathcal{D}|k} \bar{B}_k(\delta) \quad (\text{D.23i})$$

$$\sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \hat{\bar{B}}_k \leq \delta \cdot B^c y_{\text{batt}} \quad (\text{D.23j})$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \hat{\bar{S}}_k^{\text{out}} + \sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \hat{\bar{Q}}_k \geq \hat{\bar{D}}_d^Q \quad (\text{D.23k})$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} (L^Q \hat{\bar{S}}_k - \bar{S}_k(1)) + \sum_{k \in \mathcal{P}_s} \bar{s}_{(d+1)k} \bar{S}_k(1) \quad d < |\mathcal{D}|$$

$$+ \sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \hat{\bar{S}}_k^{\text{out}} \leq (E_j^Q / E_j^P) \sum_{i,k,l} \bar{p}_{ijkl} \hat{\bar{P}}_{jkl} \quad j = \text{chp} \quad (\text{D.23l})$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{1k} \bar{S}_k(1) = \sum_{k \in \mathcal{P}_s} \bar{s}_{|\mathcal{D}|k} \bar{S}_k(\delta) \quad (\text{D.23m})$$

$$\sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \hat{\bar{S}}_k^{\text{out}} \leq \delta \cdot S_{\text{stor}}^c y_{\text{stor}} \quad (\text{D.23n})$$

$$\sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \hat{\bar{Q}}_k \leq \delta \cdot S_{\text{boil}}^c y_{\text{boil}} \quad (\text{D.23o})$$

$$\bar{u}_{dk}, \quad \bar{b}_{dk}, \quad \bar{s}_{dk}, \quad \bar{q}_{dk} \in [0, 1]$$

$$\sum_{k \in \mathcal{P}_u} \bar{u}_{dk} \leq 1, \quad \sum_{k \in \mathcal{P}_b} \bar{b}_{dk} \leq 1, \quad \sum_{k \in \mathcal{P}_s} \bar{s}_{dk} \leq 1, \quad \sum_{k \in \mathcal{P}_q} \bar{q}_{dk} \leq 1. \quad (\text{D.23p})$$

References

Alguacil, N., Motto, A. L., and Conejo, A. J. (2003). Transmission expansion planning: a mixed-integer LP approach. *IEEE Transactions on Power Systems*, 18(3):1070–1077.

Allgöwer, F. and Zheng, A. (2000). *Nonlinear Model Predictive Control*, volume 26. Birkhäuser Basel.

Balas, E. (1965). An additive algorithm for solving linear programs with zero-one variables. *Operations Research*, 13(4):517–546.

Bard, J. F. (1988). Short-term scheduling of thermal-electric generators using Lagrangian relaxation. *Operations Research*, 36(5):756–766.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

Birge, J. R. (1985). Aggregation bounds in stochastic linear programming. *Mathematical Programming*, 31(1):25–41.

Birge, J. R. and Louveaux, F. (2011). *Introduction to Stochastic Programming*. Springer Science & Business Media.

Clay, R. L. and Grossmann, I. E. (1997). A disaggregation algorithm for the optimization of stochastic planning models. *Computers & Chemical Engineering*, 21(7):751–774.

Crawley, D. B., Lawrie, L. K., Winkelmann, F. C., et al. (2001). Energyplus: creating a new-generation building energy simulation program. *Energy and Buildings*, 33(4):319–331.

Crawley, D. B., Pedersen, C. O., Lawrie, L. K., and Winkelmann, F. C. (2000). Energyplus: energy simulation program. *ASHRAE Journal*, 42:49–56.

Edmunds, T. A. and Bard, J. F. (1992). An algorithm for the mixed-integer nonlinear bilevel programming problem. *Annals of Operations Research*, 34(1):149–162.

Elhallaoui, I., Villeneuve, D., Soumis, F., and Desaulniers, G. (2005). Dynamic aggregation of set-partitioning constraints in column generation. *Operations Research*, 53(4):632–645.

Falgout, R. D. (2006). An introduction to algebraic multigrid. *Computing in science & engineering*, 8:24.

Fisher, M. L. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management science*, 27(1):1–18.

Fourer, R., Gay, D. M., and Kernighan, B. W. (2012). *AMPL: A Modeling Language for Mathematical Programming*. Cengage Learning.

Garcia, C. E., Prett, D. M., and Morari, M. (1989). Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335–348.

Gelman, E. and Mandel, J. (1990). On multilevel iterative methods for optimization problems. *Mathematical Programming*, 48(1-3):1–17.

Geoffrion, A. M. (1969). An improved implicit enumeration approach for integer programming. *Operations Research*, 17(3):437–454.

Geoffrion, A. M. (1974). *Lagrangean relaxation for integer programming*. Springer.

Glover, F. (1968). Surrogate constraints. *Operations Research*, 16(4):741–749.

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8(1):156–166.

Gratton, S., Sartenaer, A., and Toint, P. L. (2008). Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization*, 19(1):414–444.

Hansen, P., Jaumard, B., and Savard, G. (1992). New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, 13(5):1194–1217.

Hedman, K. W., Ferris, M. C., O'Neill, R. P., Fisher, E. B., and Oren, S. S. (2010). Co-optimization of generation unit commitment and transmission switching with N-1 reliability. *Power Systems, IEEE Transactions on*, 25(2):1052–1063.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, 31(8):651–666.

Mathews, G. B. (1896). On the partition of numbers. *Proceedings of the London Mathematical Society*, 1(1):486–490.

Moore, J. T. and Bard, J. F. (1990). The mixed integer linear bilevel programming problem. *Operations Research*, 38(5):911–921.

Munoz, F. D., Hobbs, B., and Watson, J.-P. (2014). New bounding and decomposition approaches for MILP investment problems: Multi-area transmission and generation planning under policy constraints. Technical report, Sandia National Laboratories.

Munoz, F. D., Sauma, E. E., and Hobbs, B. F. (2013). Approximations in power transmission planning: implications for the cost and performance of renewable portfolio standards. *Journal of Regulatory Economics*, 43(3):305–338.

Pinar, A., Meza, J., Donde, V., and Lesieutre, B. (2010). Optimization strategies for the vulnerability analysis of the electric power grid. *SIAM Journal on Optimization*, 20(4):1786–1810.

Pruitt, K. A., Braun, R. J., and Newman, A. M. (2013). Evaluating shortfalls in mixed-integer programming approaches for the optimal design and dispatch of distributed generation systems. *Applied Energy*, 102:386–398.

Pruitt, K. A., Leyffer, S., Newman, A. M., and Braun, R. J. (2014). A mixed-integer nonlinear program for the optimal design and dispatch of distributed generation systems. *Optimization and Engineering*, 15:167–197.

Ren, H. and Gao, W. (2010). A MILP model for integrated plan and evaluation of distributed energy systems. *Applied Energy*, 87(3):1001–1014.

Rogers, D. F., Plante, R. D., Wong, R. T., and Evans, J. R. (1991). Aggregation and disaggregation techniques and methodology in optimization. *Operations Research*, 39(4):553–582.

Scaparra, M. P. and Church, R. L. (2008). A bilevel mixed-integer program for critical infrastructure protection planning. *Computers & Operations Research*, 35(6):1905–1923.

Siddiqui, A. S., Marnay, C., Bailey, O., and Hamachi LaCommare, K. (2005). Optimal selection of on-site generation with combined heat and power applications. *International Journal of Distributed Energy Resources*, 1(1):33–62.

Stadler, M., Marnay, C., Siddiqui, A., Lai, J., Coffey, B., and Aki, H. (2009). Effect of heat and electricity storage and reliability on microgrid viability: a study of commercial buildings in California and New York states. *Lawrence Berkeley National Laboratory Technical Report LBNL-1334E*.

Vicente, L. N. and Calamai, P. H. (1994). Bilevel and multilevel programming: A bibliography review. *Journal of Global Optimization*, 5(3):291–306.

Wen, U. P. and Hsu, S. T. (1991). Linear bi-level programming problems—a review. *Journal of the Operational Research Society*, pages 125–133.

Wen, U. P. and Huang, A. D. (1996). A simple tabu search method to solve the mixed-integer linear bilevel programming problem. *European Journal of Operational Research*, 88(3):563–571.

Wen, U. P. and Yang, Y. H. (1990). Algorithms for solving the mixed integer two-level linear programming problem. *Computers & Operations Research*, 17(2):133–142.

Wen, Z. and Goldfarb, D. (2009). A line search multigrid method for large-scale nonlinear optimization. *SIAM Journal on Optimization*, 20(3):1478–1503.

Zipkin, P. H. (1980a). Bounds for row-aggregation in linear programming. *Operations Research*, 28(4):903–916.

Zipkin, P. H. (1980b). Bounds on the effect of aggregating variables in linear programs. *Operations Research*, 28(2):403–418.